Abstract. This work proposes a model-reduction methodology that preserves Lagrangian structure (equivalently Hamiltonian structure) and achieves computational efficiency in the presence of high-order nonlinearities and arbitrary parameter dependence. As such, the resulting reduced-order model retains key properties such as energy conservation and symplectic time-evolution maps. We focus on parameterized simple mechanical systems subjected to Rayleigh damping and external forces, and consider an application to nonlinear structural dynamics. To preserve structure, the method first approximates the system’s ‘Lagrangian ingredients’—the Riemannian metric, the potential-energy function, the dissipation function, and the external force—and subsequently derives reduced-order equations of motion by applying the (forced) Euler–Lagrange equation with these quantities. From the algebraic perspective, key contributions include two efficient techniques for approximating parameterized reduced matrices while preserving symmetry and positive definiteness: matrix gappy POD and reduced-basis sparsification (RBS). Results for a parameterized truss-structure problem demonstrate the importance of preserving Lagrangian structure and illustrate the proposed method’s merits: it reduces computation time while maintaining high accuracy and stability, in contrast to existing nonlinear model-reduction techniques that do not preserve structure.

Key words. nonlinear model reduction, structure preservation, Lagrangian dynamics, Hamiltonian dynamics, structural dynamics, positive definiteness, matrix symmetry

1. Introduction. Computational modeling and simulation for simple mechanical systems characterized by a Lagrangian formalism has become indispensable across a variety of industries. Such simulations enable the understanding of complex systems, reduced design costs, and improved reliability for a wide range of applications. For example, computational structural dynamics tools have become widely used in applications ranging from aerospace to biomedical-device design; molecular-dynamics simulations have gained popularity in materials science and biology, in particular. However, the high computational cost incurred by simulating large-scale simple mechanical systems can result in simulation times on the order of weeks, even when using high-performance computers. As a result, these simulation tools are impractical for time-critical applications that demand the accuracy provided by large-scale, high-fidelity models. In particular, applications such as nondestructive evaluation for structural health monitoring, multiscale modeling, embedded control, design optimization, and uncertainty quantification require highly accurate results to be obtained quickly.

In this work, we consider models that depend on a set of parameters, e.g., design variables, operating conditions. In this context, model-reduction methods present a promising approach for addressing time-critical problems. During the offline stage, these methods perform computationally expensive ‘training’ tasks, which may include evaluating the high-fidelity model for several instances of the system parameters and computing a low-dimensional subspace for the solution. Then, during the inexpensive online stage, these methods quickly compute approximate solutions for arbitrary values of the system parameters. To accomplish this, they reduce the dimension of the high-fidelity model by restricting solutions to lie in the low-dimensional subspace that was computed offline; they also introduce other approximations when nonlinearities are present. Thus, the reduced-order model used online is characterized by a low-dimensional dynamical system that arises from a projection process on the high-fidelity-model equations. This offline/online strategy is effective primarily in two scenarios: ‘many query’ problems (e.g., Bayesian inference), where the high offline cost is amortized over many online evaluations, and real-time problems (e.g., control) characterized by stringent constraints on online evaluation time.

Generating a reduced-order model that preserves the Lagrangian structure intrinsic to mechanical systems is not a trivial task. Such structure is critical to preserve, as it leads to fundamental properties such as energy conservation (in the absence of non-conservative forces), conservation of quantities associated with symmetries in the system, and symplectic time-evolution maps. In fact, the class of structure-preserving time integrators (e.g., geometric integrators [15], variational integrators [19]) has been developed to ensure
that the discrete solution to the high-fidelity computational model associates with the time-evolution map of a (modified) Lagrangian system.

Lall et al. [18] show that performing a Galerkin projection on the Euler–Lagrange equation—as opposed to the first-order state-space form—leads to a reduced-order model that preserves Lagrangian structure. However, the computational cost of assembling the associated low-dimensional equations of motion scales with the dimension of the high-fidelity model. For this reason, this approach is efficient only when the low-dimensional operators can be assembled a priori; this occurs only in very limited cases e.g., when operators have a low-order polynomial dependence on the state and are affine in functions of the parameters [22].

Several methods have been developed in the context of nonlinear-ODE model reduction that can reduce the computational cost of assembling the low-dimensional equations of motion. However, these methods destroy Lagrangian structure when applied to simple mechanical systems. For example, collocation approaches [4, 23] perform a Galerkin projection on only a small subset of the full-order equations characterizing the high-fidelity model. Although this method works well for some nonlinear ODEs, it destroys Lagrangian structure. The discrete empirical interpolation method (DEIM) [9, 14, 11] and gappy proper orthogonal decomposition (POD) reconstruction methods [13, 6, 7] compute a few entries of the vector-valued nonlinear functions, and then approximate the uncomputed entries by interpolation or least-squares regression using an empirically derived basis. Galerkin projection can then be performed with the approximated nonlinear function. Again, this technique destroys Lagrangian structure.

The goal of this work is to devise a reduced-order model for nonlinear simple mechanical systems with general parameter dependence that leads to computationally inexpensive online solutions and preserves Lagrangian structure. We focus particularly on parameterized structural-dynamics models under Rayleigh damping and external forces. The methodology we propose constructs a reduced-order model by first approximating the ‘Lagrangian ingredients’ (i.e., quantities defining the problem’s Lagrangian structure) and subsequently deriving the equations of motion by applying the Euler–Lagrange equation to these ingredients. The method approximates the Lagrangian ingredients as follows:

I. **Configuration space.** The low-dimensional configuration space is derived using standard dimension-reduction techniques, e.g., proper orthogonal decomposition, modal decomposition.

II. **Riemannian metric.** The Riemannian metric is defined by a low-dimensional symmetric positive-definite matrix. We propose two efficient methods for approximating this low-dimensional matrix that preserve symmetry and positive definiteness.

III. **Potential-energy function.** The potential energy function is approximated by employing the original potential-energy function, but with the low-dimensional reduced-basis matrix replaced by a low-dimensional sparse matrix with only a few nonzero rows. This sparse matrix is computed online by matching the gradient of the reduced potential to first order about the equilibrium configuration.

IV. **Dissipation function.** The damping matrix associated with the Rayleigh dissipation function is a linear combination of the mass matrix (which defines the Riemannian metric) and the Hessian of the potential. Thus, we form the approximated Rayleigh dissipation function in the same fashion, but employ the approximated mass matrix from ingredient [II] and approximated potential from ingredient [III].

V. **External force.** The external force is derived by applying the Lagrange–D’Alembert principle with variations in the configuration space. We approximate this by applying gappy POD reconstruction to the external force as expressed in the original coordinates. As a result, the external force appearing in the reduced-order equations of motion can be derived by applying the Lagrange–D’Alembert principle to this modified external force with variations restricted to the low-order configuration space.

We note that a structure-preserving method [5] has been recently proposed for nonlinear port-Hamiltonian systems, which are generalizations of Hamiltonian systems. While this technique guarantees that properties such as stability and passivity are preserved, it does not in fact preserve Lagrangian or classical Hamiltonian structure. In particular, the resulting reduced-order equations of motion cannot be derived from approximated ingredients such as those enumerated above; as a consequence, the approach does not ensure symplecticity or energy conservation for conservative systems, for example.

As hinted above, preserving structure for Lagrangian ingredient [II] is equivalent to efficiently approximating a low-dimensional reduced matrix while preserving symmetry and positive definiteness. This algebraic task is relevant to a broad scope of applications, e.g., approximating extreme eigenvalues/eigenvectors of a parameterized matrix, preserving Hessian positive definiteness in optimization algorithms. For this reason,
Section 2 presents approximation techniques for Lagrangian ingredient I in a stand-alone algebraic setting that does not rely on the Lagrangian formalism. Similarly, Section 3 considers Lagrangian ingredient II in a purely algebraic context that does not depend on Lagrangian dynamics.

The remainder of the paper is organized as follows. Section 4 introduces the Lagrangian-mechanics formulation. Section 5 outlines existing model-reduction techniques and highlights the need for an efficient, structure-preserving method. Section 6 presents the proposed method. Section 7 presents numerical experiments applied to a simple mechanical system from structural dynamics. Finally, Section 8 summarizes the contributions and suggests further research.

The structure-preserving model-reduction methods proposed in this work also preserve Hamiltonian structure when the Hamiltonian formulation of classical mechanics is used. Appendix A provides this connection.

2. Preserving matrix symmetry and positive definiteness. This section presents approximation techniques for Lagrangian ingredient I in an algebraic setting. First, to establish notation, denote the system connections when the Hamiltonian formulation of classical mechanics is used. Appendix A provides this connection.

(P1) At a cost independent of \( N \), compute a symmetric positive-definite matrix \( \tilde{A}(\mu^*) \) that is appropriately close to the matrix \( V^T A(\mu^*) V \) for any specified online point \( \mu^* \in \mathcal{D} \).

Note that due to the density of \( V \), directly computing \( V^T A(\mu^*) V \) requires computing all \( \mathcal{O}(N) \) entries of the matrix \( A(\mu^*) \). Recall that the offline/online strategy we adopt permits expensive offline operations that facilitate the solution to online problem (P1). These operations may include collecting \( p \) ‘snapshots’ of the matrix \( A(\mu^i), i = 1, \ldots, p \), where \( \mu^i \in \mathcal{D} \) denotes the \( i \)th instance of the training set.

We assume that computing a single entry of \( A(\mu^*) \) for any specified online point \( \mu^* \in \mathcal{D} \) is inexpensive, i.e., the number of floating-point operations (flops) is independent of \( N \). However, we make no other assumptions regarding the parameters or the matrix. In particular, we do not assume affine parametric dependence of the matrix, and we view \( \mu \mapsto A(\mu) \) simply as a mechanism for generating instances of the matrix \( A \).

We now present two methods for solving online problem (P1). Method 1 approximates the reduced matrix by projecting the full matrix onto a sparse basis, while Method 2 approximates the reduced matrix as a linear combination of pre-computed reduced matrices. Later, Section 5.2 contrasts the proposed methods with existing model reduction approaches such as DEIM, gappy POD, and collocation. These existing methods apply a one-sided sampling operator to the matrix, i.e., they replace \( V^T \) with some type of sparse matrix. While this leads to an inexpensive approximation, it gives rise to a non-symmetric reduced matrix approximation. This destroys the underlying problem structure and therefore fails to meet the requirements of online problem (P1).

2.1. Reduced-basis sparsification (RBS). We first consider a strategy that ‘injects sparseness’ into the matrix \( V \). That is, we replace \( V \) by \( U_A \in \mathbb{R}^{N \times n} \), which has full column rank and only \( m \) rows (with \( n \leq m \ll N \)) containing nonzero entries:

\[
\tilde{A}(\mu) = U_A^T A(\mu) U_A.
\] (2.1)

This sparse matrix may be expressed as \( U_A \equiv PU_A \), where \( P \in \{0,1\}^{N \times m} \) is a ‘sampling matrix’ consisting of \( m \) selected columns of the \( N \times N \) identity matrix, \( U_A \in \mathbb{R}^{m \times n} \) is a dense matrix with full column rank \( n \), and \( \mathbb{R}^{m \times n} \) denotes the noncompact Stiefel manifold: the set of full-rank \( m \times n \) matrices. Clearly, \( \tilde{A}(\mu) \) is symmetric positive definite if \( A(\mu) \) is symmetric positive definite; thus, the approximation defined by (2.1) preserves the requisite structure. Note that this approximation will also preserve structure in cases where \( A(\mu) \) is symmetric positive semidefinite or simply symmetric. Further, the (online) operation count for computing \( \tilde{A}(\mu^*) \) for online point \( \mu^* \in \mathcal{D} \) is independent of \( N \). Computing \( P^T A(\mu^*) P \) is equivalent to computing only \( m^2 \) (symmetric) entries of \( A(\mu^*) \) and entails \( \mathcal{O}(m^2) \) flops; subsequently computing \( \tilde{A}(\mu^*) = U_A^T [P^T A(\mu^*) P] U_A \) requires \( \mathcal{O}(m^2 n + mn^2) \) flops.
Given a sampling matrix $P$, the matrix $U_A$ can be computed offline to minimize the average approximation error over the snapshots, i.e., according to the following optimization problem:

$$U_A = \arg \min_{X \in \mathbb{R}^{m \times n}} \sum_{i=1}^{p} \| X^T P^T A (\mu_i) P X - V^T A (\mu_i) V \|_F^2,$$  \hspace{1cm} (2.2)

where the subscript $F$ denotes the Frobenius norm. To handle the fact that $\mathbb{R}^{m \times n}$ is an open set, optimization problem (2.2) can first be solved over $\mathbb{R}^{m \times n}$ and the solution can be subsequently projected onto $\mathbb{R}^{m \times n}$, which is analogous to the approach taken by Vandereycken [25] Algorithm 6. We note that other objective functions may be considered for specialized online analyses, e.g., the approximation error of extreme eigenvalues or eigenvectors over the matrix snapshots. Note that problem (2.2) is a small-scale optimization problem, as $m,n \ll N$. It can be solved at a cost independent of $N$ (for each optimization iteration) during the offline stage after the matrix snapshots $A (\mu_i), i = 1, \ldots, p$ and their reduced counterparts $V^T A (\mu_i) V, i = 1, \ldots, p$ have been computed.

Proceedure 1 provides the offline and online steps required to implement the RBS approximation.

**Procedure 1: Reduced-basis sparsification for symmetric matrices**

**Offline stage**

1. Collect matrix snapshots $A (\mu_i), i = 1, \ldots, p$.
2. Form reduced matrices $V^T A (\mu_i) V, i = 1, \ldots, p$.
3. Choose the sample matrix $P$.
4. Determine $U_A$ as the solution to problem (2.2).

**Online stage** (given $\mu^*$)

1. Compute $P^T A (\mu^*) P$.
2. Form $\tilde{A} (\mu^*) = U_A^T [P^T A (\mu^*) P] U_A$.

**Remark.** This paper does not focus on methods for selecting the sampling matrix $P$, a task that is typically carried out during the offline stage and uses the collected snapshots. All numerical experiments presented in Section 7 use the GNAT greedy approach [6] for this purpose. This method has proven to be quite robust, even when applied to the approximation techniques proposed in this paper. A more careful study of sampling algorithms will be addressed in future work, where ideas of tailoring the sampling approach to the specific reduced-order modeling approximation will be explored.

2.1.1. Exactness conditions. In the full-sampling case where $m = N$, the approximation is exact if problem (2.2) is solved via a gradient-based method with an initial guess of $X^{(0)} = P^T V$; we do this in practice. Under these conditions, $U_A = V$ and so $\tilde{A} (\mu) = V^T A (\mu) V$.

In the general case where $m < N$, it is possible to show that the approximation is exact if the matrix is parameter-independent (i.e., $A (\mu) = A$) and $m \geq n$. This situation is considered in the discussion that follows Theorem 2.1 below. It is also possible to prove a more general exactness result in cases where the parametric dependence of $A (\mu)$ is relatively simple. In particular, consider the parametric form

$$A (\mu) = h_1 (\mu) A_1 + h_2 (\mu) A_2,$$  \hspace{1cm} (2.3)

where $A_1$ and $A_2$ are $N \times N$ symmetric positive-definite matrices and $h_1,h_2 : \mathcal{D} \rightarrow \mathbb{R}$. It can then be shown that a sparse reduced basis exists that exactly captures $V^T A (\mu) V$ under conditions related to how well the eigenvalues of the sampled matrix $P^T A (\mu) P$ encompass (or surround) those of the reduced matrix $V^T A (\mu) V$. Loosely stated, the encompassing conditions amount to how well the sampled matrix captures the behavior of the reduced matrix. Formally, the following theorem makes precise the notion of encompassing using eigenvalue interlacing ideas from classical linear algebra.

**Theorem 2.1.** Let $A (\mu)$ have the form given by Eq. (2.3).
Then,
\[ \exists \mathbf{U}_\mathbf{A} \in \mathbb{R}^{m \times n} \text{ such that } \mathbf{U}_\mathbf{A}^T \mathbf{P}^T \mathbf{A} (\mu) \mathbf{P} \mathbf{U}_\mathbf{A} = \mathbf{V}^T \mathbf{A} (\mu) \mathbf{V}, \quad \forall \mu \in \mathcal{D} \] (2.4)
if and only if the generalized eigenvalues of \((\mathbf{V}^T \mathbf{A}_2 \mathbf{V}, \mathbf{V}^T \mathbf{A}_1 \mathbf{V})\) interlace the generalized eigenvalues of \((\mathbf{P}^T \mathbf{A}_2 \mathbf{P}, \mathbf{P}^T \mathbf{A}_1 \mathbf{P})\), i.e.,
\[ \lambda_i^{(s)} \leq \lambda_i^{(r)} \leq \lambda_{i+m-n}^{(s)} \quad \text{for } i = 1, \ldots, n, \] (2.5)
with
\[ [\mathbf{V}^T \mathbf{A}_2 \mathbf{V}] \mathbf{x}_i^{(r)} = \lambda_i^{(r)} [\mathbf{V}^T \mathbf{A}_1 \mathbf{V}] \mathbf{x}_i^{(r)}, \quad i = 1, \ldots, n \]

and
\[ [\mathbf{P}^T \mathbf{A}_2 \mathbf{P}] \mathbf{x}_i^{(s)} = \lambda_i^{(s)} [\mathbf{P}^T \mathbf{A}_1 \mathbf{P}] \mathbf{x}_i^{(s)}, \quad i = 1, \ldots, m. \]

Note that the eigenvalues are sorted in order of increasing magnitude. Appendix D.1 contains the proof.

Here, we discuss the theorem’s implications.

When \( \mathbf{A} \) is independent of \( \mu \), we can choose \( h_1 = h_2 = 1 \) and \( \mathbf{A}_1 = \mathbf{A}_2 \). The interlacing property is then trivially satisfied for \( m = n \) with \( \lambda_i^{(s)} = \lambda_i^{(r)} = 1 \), and so the equality in (2.4) always holds. When instead \( \mathbf{A}_1 \neq \mathbf{A}_2 \) and \( m = n+1 \), the interlacing definition is quite restrictive, as it implies that \( \lambda_k^{(s)} \leq \lambda_k^{(r)} \leq \lambda_{k+1}^{(s)} \).

We would not generally expect the eigenvalues of the sampled and reduced matrices to have this property. However, when \( m \gg n+1 \), each interval width is (much) larger and so the condition is not nearly as restrictive. For example, if \( n = 100 \) and \( m = 300 \), then interlacing implies that \( \lambda_i^{(s)} \leq \lambda_i^{(r)} \leq \lambda_{i+200}^{(s)} \). Thus, we generally expect the conditions of the theorem to be satisfied for sufficiently large \( m \), though this is not guaranteed and depends on matrix spectrum. As a final note, although the theorem assumes a specific form of \( \mathbf{A} (\mu) \), it should characterize the rough behavior of a more general \( \mathbf{A} (\mu) \) that does not vary ‘too much’ from the affine functional form (2.3).

2.2. Matrix gappy POD. An alternative structure-preserving approximation applicable to problem (1.1) assumes the following form:
\[ \mathbf{\hat{A}} (\mu) = \sum_{i=1}^{nA} \xi_i^A (\mu) \mathbf{V}^T \mathbf{A}_i \mathbf{V}. \] (2.6)

Here, the matrices \( \mathbf{A}_i, \ i = 1, \ldots, n_A \) are \( N \times N \) symmetric matrices that are computed offline and define a basis for the matrix \( \mathbf{A} (\mu) \). Due to the symmetry of \( \mathbf{A}_i, \ i = 1, \ldots, n_A \), the approximation \( \mathbf{\hat{A}} (\mu) \) will always be symmetric. The parameter-dependent coefficients \( \xi_A \equiv (\xi_1^A, \ldots, \xi_{nA}^A) \) are computed online in an efficient manner that ensures \( \mathbf{\hat{A}} (\mu) \) is positive definite and thereby preserves requisite structure.

The next sections describe procedures for computing the matrix basis and coefficients. We refer to this method as ‘matrix gappy POD’, as it amounts to the gappy POD procedure [13] applied to matrix data with modifications to preserve positive definiteness. The approach, which we originally proposed [8], is a more general formulation of the ‘matrix DEIM’ approach [26] (or ‘multi-component EIM’ [24] in the context of the reduced-basis method applied to parametrized non-affine elliptic PDEs), as it permits least-squares reconstruction (not simply interpolation). Further, it is equipped with a mechanism to maintain positive definiteness.

2.2.1. Offline computation: matrix basis. To obtain the matrix basis, we propose applying a vectorized POD method, wherein the basis can be considered a set of ‘principal matrices’ that optimally represent the matrix \( \mathbf{A} \) over the training set \( \{ \mu^j \} \). The (offline) steps for this method are as follows:

1. Collect matrix snapshots \( \mathbf{A} (\mu^j), \ i = 1, \ldots, p. \)

These matrices are optimal in the sense that they minimize the average projection error (as measured in the Frobenius norm) of the matrix snapshots.
2. Vectorize the snapshots \( a^i \equiv v(A(\mu^i)) \in \mathbb{R}^{N^2}, i = 1, \ldots, p \), where the function \( v : \mathbb{R}^{N \times N} \rightarrow \mathbb{R}^{N^2} \) vectorizes a matrix.

3. Compute an \( n_A \)-dimensional (with \( n_A \leq p \)) POD basis of the vectorized snapshots

\[
W_a \equiv [a^1 \cdots a^{n_A}] \in \mathbb{R}^{N^2 \times n_A}
\]  

(2.7)

using vectorized snapshots \( \{a^i\}_{i=1}^p \) and an ‘energy criterion’ \( \eta_A \in [0, 1] \) as inputs to Algorithm 1 of Appendix B.

4. Transform these POD basis vectors into their matrix counterparts:

\[
A_i = v^{-1}(a^i), \quad i = 1, \ldots, n_A.
\]  

(2.8)

Each matrix \( A_i \), \( i = 1, \ldots, n_A \) is guaranteed to be symmetric, as Algorithm 1 forms this basis by taking linear combinations of symmetric matrices.

### 2.2.2. Online computation: coefficients.

The approximation error can be bounded as follows:

\[
\|V^T A(\mu) V - \hat{A}(\mu)\|_F = \|V^T A(\mu) V - \sum_{i=1}^{n_A} \xi_A^i (\mu) V^T A_i V\|_F 
\]

\[
\leq \|V\|_F^2 \|A(\mu) - \sum_{i=1}^{n_A} \xi_A^i (\mu) A_i\|_F
\]  

(2.9)

(2.10)

where \( \|V\|_F^2 = n \) if \( V \) is orthogonal. This leads to a natural choice for the scalar coefficients based on minimizing the upper bound \( (2.10) \). In particular, we compute coefficients \( \xi_A(\mu^*) \) online for a specific \( \mu^* \in \mathcal{D} \) as the solution to

\[
\begin{align*}
\text{minimize} & \quad \|P^T A(\mu^*) P - \sum_{i=1}^{n_A} x_i P^T A_i P\|_F^2 \\
\text{subject to} & \quad \sum_{i=1}^{n_A} x_i V^T A_i V > 0.
\end{align*}
\]

(2.11)

Note that the coefficients are computed to match (as closely as possible) the full matrix and the linear combination of pre-computed full matrices at a few entries. The constraints amount to a strict linear-matrix-inequality, where \( A > 0 \) denotes a generalized inequality that indicates \( A \) is a positive-definite matrix. This constraint ensures that structure is preserved. Note that the constraint can be modified (resp. dropped) in cases where positive semidefiniteness (resp. simply symmetry) aims to be preserved.

Problem (2.11) is equivalent to a linear least-squares problem with nonlinear constraints; this can be seen from its vectorized form:

\[
\begin{align*}
\text{minimize} & \quad \|\hat{P}^T v(A(\mu^*)) - \hat{P}^T W_a x\|_2^2 \\
\text{subject to} & \quad \sum_{i=1}^{n_A} x_i V^T A_i V > 0.
\end{align*}
\]

(2.12)

Here, \( \hat{P} \) is an alternate form of the sampling matrix that can be applied to vectorized matrices, i.e. \( \hat{P}^T v(A(\mu^*)) = v(P^T A(\mu^*) P) \).

The objective function is equivalent to that of the gappy POD method [13]—which will be further discussed in Section 5.2.2—applied to matrix data. Note that this optimization problem is solved online

2Exploiting symmetry, this sampling matrix can be expressed as \( \hat{P} \equiv [p^1 \cdots p^{(m^2 + m)/2}] \in \{0,1\}^{N^2 \times (m^2 + m)/2} \) where \( p^{i+(j^2-j)/2} = v(p^i [p^j]^T) \) for \( i = 1, \ldots, j \) and \( j = 1, \ldots, m \) and \( P \equiv [p^1 \cdots p^m] \). From the definition of \( p^{i+(j^2-j)/2} \), it follows that \( p^{i+(j^2-j)/2} \) extracts the \( (i,j) \)th entry from the vectorized form of a matrix.
using the online-sampled data $P^T A(\mu^*) P$; Appendix C describes a method for solving this optimization problem. In practice, we usually observe the constraints to be inactive at the unconstrained solution. Therefore, typically the constraints need not be handled directly, and solving problem (2.11) amounts to solving a small-scale linear least-squares problem characterized by an $(m^2 + m)/2 \times n_A$ matrix. To ensure a unique solution to problem (2.11), the matrix $P^T W_a$ must have full column rank. This can be achieved by enforcing $(m^2 + m)/2 \geq n_A$ as well as mild conditions on the sampling matrix $P$.

Procedure 2 describes the offline and online stages for implementing the matrix gappy POD approximation.

**Procedure 2: Matrix gappy POD**

**Offline stage**

1. Compute the basis matrices $A_i$, $i = 1, \ldots, n_A$ using the vectorized POD approach described in Section 2.2.1.
2. Determine the sampling matrix $P$ which gives rise to a full column rank matrix $P^T W_a$ and with $m$ chosen so that $(m^2 + m)/2 \geq n_A$.
3. Compute low-dimensional matrices $V_i A_i V_i$, $i = 1, \ldots, n_A$.
4. Retain the sampled entries of the matrix basis $P^T A_i P$, $i = 1, \ldots, n_A$; discard other entries.

**Online stage** (given $\mu^*$)

1. Compute $P^T A(\mu^*) P$.
2. Solve the small-scale optimization problem (2.11) for coefficients $\xi^T (\mu^*)$.
3. Assemble the low-dimensional matrix $\tilde{A} (\mu^*)$ by Eq. (2.6).

### 2.2.3. Exactness conditions. Theorem 2.2.

The matrix gappy POD approximation is exact for any specified online parameters $\mu^* \in D$ if

1. $v(A(\mu^*)) \in \text{range}(W_a)$ and
2. $P^T W_a$ has full column rank.

See Appendix D.2 for the proof.

Condition 1 holds, e.g., when $\mu^* \in \{\mu^i\}$ and $n_A = p$. Condition 2 can be straightforwardly enforced by the choice of $P$ and automatically holds in the case of full sampling, i.e., $m = N$.

### 3. Preserving potential-energy structure.

This section presents a technique for approximating Lagrangian ingredient III within an algebraic setting. To begin, define a parameterized scalar-valued function $V : \mathbb{R}^N \times D \to \mathbb{R}$ with $(q, \mu) \mapsto V$ that is nonlinear in both arguments and can be interpreted as a Lagrangian dynamical system’s (parameterized) potential energy. Here, $q \in \mathbb{R}^N$ denotes the system’s configuration variables and $\mu \in D$ denotes the system parameters that belong to parameter domain $D$. Unlike the matrix approximations of the previous section, the nonlinear dependence on the configuration variables $q$ introduces additional challenges that must be considered carefully.

We aim to devise an offline method—which may entail expensive operations—for constructing a scalar-valued function $\tilde{V}_c : \mathbb{R}^n \times D \to \mathbb{R}$. This function will be used online and should satisfy the demands of online problem (P2):

(P2) Compute the gradient vector $\nabla_q \tilde{V}_c(q_c^*; \mu^*)$ at a cost independent of $N$. Given any online parameter $\mu^* \in D$, this vector should be appropriately close to $V^T \nabla_q V(q(\mu^*) + \nabla q_c^*; \mu^*)$ for all coordinates $q_c^* \in \mathbb{R}^n$.

As before, $V$ represents a dense, parameter-independent, full-column-rank $N \times n$ matrix. We denote by $\tilde{g} : D \to \mathbb{R}^N$ a parameterized reference configuration about which the low-dimensional reduced configuration space is centered. Notice that this problem is concerned with approximating the gradient of the scalar-valued function as opposed to the function itself. As will be discussed in Section 5, this problem arises in model reduction of parameterized Lagrangian-dynamics systems, where the gradient of the potential appears in the equations of motion.
In certain specialized cases, the above approximation can be simplified considerably. For example, when \( \tilde{q}(\mu) = 0, \forall \mu \in \mathcal{D} \) and the function \( V(q; \mu) \) is purely quadratic in its first argument, then \( V^T \nabla_q V(\tilde{q}(\mu) + V q_r; \mu) = V^T A(\mu) V q_r \), where \( A(\mu) \) is a symmetric Hessian matrix; in this case, one of the approximation techniques described in Section 2 can be straightforwardly applied. Alternatively, if the potential energy is defined by the integral over a domain (i.e., \( V(q; \mu) = \int_{\Omega} V(X, q; \mu) d\Omega_X \)), a sparse cubature method \( \mathbb{C} \) can be used to achieve computational efficiency and structure preservation. In more general cases, however, another approach is needed. In the following, we develop a method that makes no simplifying assumptions about the dependence of the potential \( V \) on the configuration variables or parameters.

Due to the density of the matrix \( V \), the most straightforward approach of setting \( \tilde{V}_r(q_r; \mu) = V(\tilde{q}(\mu) + V q_r; \mu) \) leads to expensive online operations: computing the gradient \( \nabla_q \tilde{V}_r(q_r^*; \mu^*) = V^T \nabla_q V(\tilde{q}(\mu^*) + V q_r^*; \mu^*) \) requires first computing all \( N \) entries of the gradient vector \( \nabla_q V(\tilde{q}(\mu^*) + V q_r^*; \mu^*) \). To rectify this, we revisit the RBS technique proposed in Section 2.1 and introduce some minor modifications. In particular, we replace \( V \) by a sparse parameter-dependent matrix \( U_V(\mu) \equiv \mathbb{P} U_V(\mu) \in \mathbb{R}^{N \times n} \) with only \( m \ll N \) nonzero rows, where \( U_V(\mu) \in \mathbb{R}^{m \times n} \) is a dense matrix. That is, we approximate the potential energy as

\[
\tilde{V}_r(q_r; \mu) = V(\tilde{q}(\mu) + U_V(\mu) q_r; \mu). \tag{3.1}
\]

This approximation preserves structure, as \( \tilde{V}_r \) remains a parameterized scalar-valued function. Now, we wish to compute \( U_V \) such that \( \nabla_q \tilde{V}_r(q_r^*; \mu^*) = U_V(\mu^*)^T \nabla_q V(\tilde{q}(\mu^*) + U_V(\mu^*) q_r^*; \mu^*) \) is as close as possible to \( V^T \nabla_q V(\tilde{q}(\mu^*) + V q_r^*; \mu^*) \) for any online point \( \mu \in \mathcal{D} \) and any \( q_r^* \in \mathbb{R}^n \). One can imagine a variety of methods for computing \( U_V \) toward this stated goal. For example, one can formulate an optimization problem to match the potential gradient at training points \( \mathbb{S} \); this effectively leads to a parameter-independent matrix \( U_V \). However, we found this approach to lead to significant errors for many problems. Instead, we pursue an idea motivated by the analysis in Section 3.1 which centers on the first two terms in a Taylor expansion of \( V^T \nabla_q V(\tilde{q}(\mu^*) + V q_r^*; \mu^*) \) about the reference configuration.

In practice, we often find that the trajectories of dynamical systems are localized in the configuration space. This is particularly true for mechanical oscillators often encountered in structural dynamics, where the trajectory does not deviate drastically from the equilibrium configuration. Using this observation, we focus our approximation efforts on accurately capturing the behavior of the potential in a neighborhood of the online reference configuration \( \tilde{q}(\mu^*) \). Implicitly, this assumes that the online configurations do not greatly diverge from this point. To this end, consider computing \( U_V(\mu^*) \) online such that the approximation \( U_V(\mu^*)^T \nabla_q V(\tilde{q}(\mu^*) + U_V(\mu^*) q_r^*; \mu^*) \) matches \( V^T \nabla_q V(\tilde{q}(\mu^*) + V q_r^*; \mu^*) \) to first order about the reference configuration:

\[
U_V(\mu^*)^T \nabla_q V(\tilde{q}(\mu^*) + U_V(\mu^*) q_r^*; \mu^*) \approx V^T \nabla_q V(\tilde{q}(\mu^*) + V q_r^*; \mu^*) \quad \forall q_r^* \in \mathbb{R}^n.
\tag{3.2}
\]

Notice that the high-order terms amount to approximating a reduced Hessian (defined via the dense matrix \( V \)) by a second reduced Hessian (defined via the sparse matrix \( U_V(\mu^*) \)). This is equivalent to online problem (I) presented in Section 2 that was addressed by the RBS algorithm (as well as a matrix gappy POD approach). This RBS algorithm is supported by Theorem 2.1 which shows that an exact approximation of the reduced Hessian is possible under certain assumptions. While these assumptions do not always hold, the theorem gives an expectation that a good approximation can be found under more general circumstances. Unfortunately, the presence of the low-order terms in Eq. (3.2) alters the character of the reduced approximation and so Theorem 2.1 no longer applies. In this case, the matrix \( U_V(\mu^*) \) must serve to capture both gradient and Hessian information simultaneously, which introduces restrictive assumptions in order to obtain an equivalent result to Theorem 2.1; this will be shown in Lemma 2.1.

To avoid the limitations associated with these restrictions, we choose the reference configuration to be equilibrium, i.e., \( \tilde{q}(\mu) = q_0(\mu) \) with equilibrium defined as \( \nabla_q V(q_0(\mu); \mu) = 0 \). This forces the low-order Taylor terms to zero and simplifies Eq. (3.2) to

\[
U_V(\mu^*)^T \nabla_{qq} V(q_0(\mu); \mu^*) U_V(\mu^*) = V^T \nabla_{qq} V(q_0(\mu); \mu^*) V. \tag{3.3}
\]
Now, Theorem 2.1 holds, implying that Equation (3.3) can be exactly solved when \( m = n \). For this reason, we compute \( \mathbf{U}_V(\mu^*) \) online for each \( \mu^* \) to satisfy (3.3) using \( n \) sample indices. Specifically, we define it according to

\[
\mathbf{U}_V(\mu^*) = \begin{bmatrix} \mathbf{X} \\ 0_{(m-n) \times n} \end{bmatrix},
\]

(3.4)

where \( \mathbf{X} \) is given by solving

\[
\mathbf{L}_1^T \mathbf{X} = \mathbf{L}_2^T,
\]

\( \mathbf{L}_2 \in \mathbb{R}^{n \times n} \) denotes the lower-triangular Cholesky factor of \( \mathbf{V}^T \nabla_{\mathbf{q}_0} V(\mathbf{q}_0(\mu^*); \mu^*) \mathbf{V}, \mathbf{L}_1 \in \mathbb{R}^{n \times n} \) denotes the lower-triangular Cholesky factor of \( \mathbf{P}_1^T \nabla_{\mathbf{q}_0} V(\mathbf{q}_0(\mu^*); \mu^*) \mathbf{P}_1 \), and \( \mathbf{P}_1 \) represents the first \( n \) columns of \( \mathbf{P} \). We defer discussing the computational cost for this approach to Section 3.2 and now return to the previously alluded difficulties associated with solving (3.2) when the reference configuration does not correspond to equilibrium.

3.1. Solvability of the two-term Taylor equation. The method presented in the previous section was motivated by difficulties in inexpensively approximating the reduced gradient of a nonlinear function. In this section, we give some insight into these difficulties by investigating a much easier situation: the solvability of the two-term Taylor equation (3.2), which we write in matrix/vector form as

\[
\mathbf{U}_V^T \mathbf{P}^T \mathbf{c} + \mathbf{U}_V^T \mathbf{P}^T \mathbf{A} \mathbf{U}_V \mathbf{q}_* = \mathbf{V}^T \mathbf{c} + \mathbf{V}^T \mathbf{A} \mathbf{q}_*, \quad \forall \mathbf{q}_* \in \mathbb{R}^n.
\]

(3.5)

Here, we have set \( \mathbf{c} = \nabla_{\mathbf{q}} V(\tilde{\mathbf{q}}(\mu^*); \mu^*) \) and \( \mathbf{A} = \nabla_{\mathbf{q}_0} V(\tilde{\mathbf{q}}(\mu^*); \mu^*) \). We have also dropped dependence on \( \mu^* \) such that \( \mathbf{c} \) and \( \mathbf{A} \) are parameter independent in the following analysis; this is equivalent to restricting equation (3.2) to a single instance of \( \mu^* \). This is somewhat less than ideal in that we would normally wish to minimize online costs by computing a single \( \mathbf{U}_V \) during the offline phase that is then valid for all subsequent online calculations. However, what we now show is that it is not always possible to satisfy equation (3.2) even when one is restricted to finding a \( \mathbf{U}_V \) for a single instance of \( \mu^* \).

As (3.5) must hold for all \( \mathbf{q}_* \), we have the following two necessary and sufficient conditions

\[
\mathbf{U}_V^T \mathbf{P}^T \mathbf{A} \mathbf{U}_V = \mathbf{V}^T \mathbf{A} \mathbf{V} \quad \text{and} \quad \mathbf{U}_V^T \mathbf{P}^T \mathbf{c} = \mathbf{V}^T \mathbf{c}.
\]

(3.6)

It is possible to show that satisfying these conditions is equivalent to finding a \( \mathbf{\bar{U}}_V \in \mathbb{R}^{m \times n} \) such that

\[
\mathbf{\bar{U}}_V^T \mathbf{\bar{U}}_V = \mathbf{I} \quad \text{and} \quad \mathbf{\bar{U}}_V^T \mathbf{P}^T \mathbf{c} = \mathbf{\bar{V}}^T \mathbf{c}
\]

(3.7)

where

\[
\mathbf{\bar{V}}^T \mathbf{\bar{V}} = \mathbf{I}.
\]

The definitions of \( \mathbf{\bar{U}}_V, \mathbf{\bar{V}}, \) and \( \mathbf{\bar{c}} \) are given below. The key point is that the necessary and sufficient conditions for equation (3.7) amount to finding an orthogonal matrix, \( \mathbf{\bar{U}}_V \), such that \( \mathbf{\bar{U}}_V^T \mathbf{P}^T \mathbf{c} = \mathbf{\bar{V}}^T \mathbf{c} \) for a given orthogonal matrix \( \mathbf{\bar{V}} \), and a given vector, \( \mathbf{\bar{c}} \). In the simple case when \( \mathbf{A} \) is the identity and \( \mathbf{V} \) is orthogonal, we have \( \mathbf{\bar{U}}_V = \mathbf{U}_V \) and \( \mathbf{\bar{V}} = \mathbf{V} \). More generally, we have the following definitions:

\[
\mathbf{\bar{U}}_V = \mathbf{P}^T \mathbf{L}^T \mathbf{P} \mathbf{U}_V \mathbf{L}_\phi^{-T}, \quad \mathbf{\bar{V}} = \mathbf{L}^T \mathbf{V} \mathbf{L}_\phi^{-T}, \quad \mathbf{\bar{c}} = \mathbf{L}^{-1} \mathbf{c},
\]

where \( \mathbf{L} \) is the lower-triangular Cholesky factor of \( \mathbf{A} \), and \( \mathbf{L}_\phi \) is the lower-triangular Cholesky factor of \( \mathbf{V}^T \mathbf{A} \mathbf{V} \). The above equivalence hinges on the identities \( \mathbf{P}^T \mathbf{L}^T \mathbf{P} = \mathbf{P}^T \mathbf{A} \mathbf{P} \) and \( \mathbf{P}^T \mathbf{L}^T \mathbf{P} \mathbf{L}^{-1} = \mathbf{P}^T \). These hold due to the lower-triangular form of the matrix \( \mathbf{L} \).

The following lemma addresses the conditions under which Eq. (3.7) or equivalently Eq. (3.5) hold.

**Lemma 1.** Consider the equations

\[
\mathbf{\bar{U}}_V^T \mathbf{\bar{U}}_V = \mathbf{I} \quad \text{and} \quad \mathbf{\bar{U}}_V^T \mathbf{P}^T \mathbf{c} = \mathbf{\bar{V}}^T \mathbf{c}
\]

(3.8)
with the following matrices given: \( P \in \{0, 1\}^{N \times n} \) consists of selected columns of the identity matrix (see prior definition), \( V \in \mathbb{R}^{N \times n} \) with \( V^T V = I \), and \( \hat{c} \in \mathbb{R}^{N \times 1} \). Then, assuming that \( m \geq n \), some \( \hat{U}_V \in \mathbb{R}^{m \times n} \) exists such that Eq. (3.8) is satisfied if and only if
\[
||V^T \hat{c}||_2 = ||P^T \hat{c}||_2 \text{ and } m = n \quad \text{or} \quad ||V^T \hat{c}||_2 \leq ||P^T \hat{c}||_2 \text{ and } m > n.
\]

See Appendix D.3 for the proof.

Obviously, equation (3.9) is satisfied if either \( \hat{c} = 0 \) (i.e., the equilibrium configuration is taken as the reference configuration) or if \( m = N \). Unfortunately, however, equation (3.9) is not guaranteed to be satisfactory in more general situations. Specifically, when \( m > n \), the condition \( ||V^T \hat{c}||_2 \leq ||P^T \hat{c}||_2 \) corresponds to comparing the magnitude of a vector of length \( n \) obtained by rotating and dropping components with a second vector of length \( m \) obtained by simply dropping components. If \( V \) and \( \hat{c} \) are not correlated, then one could perhaps hope that on average the vector with more components would generally have a larger magnitude. However, when \( \hat{c} \) lies completely within the subspace spanned by the columns of \( V \) and all components of \( \hat{c} \) are non-zero, then \( ||V^T \hat{c}||_2 = ||\hat{c}||_2 \) and so satisfying the necessary and sufficient conditions requires ‘full sampling’ \( m = N \). While this scenario may be considered pessimistic, one can expect that a very large value of \( m \) will be required when \( \hat{c} \) lies primarily in the range space of \( V \). In general, there is no guarantee that even the simplified (i.e., parameter-independent) form of the two-term Taylor equation is solvable. When one also considers that Eq. (3.5) corresponds to the restriction of Eq. (3.2) to a single instance of \( \mu^* \), the above result should be seen as quite discouraging.

For this reason, we abandon any attempt at computing a parameter-independent sparse matrix \( U_V \) during the offline phase that can serve to approximate the reduced gradient for all online points \( \mu^* \in \mathcal{D} \). Instead, we limit ourselves to the online computation of a parameter-dependent matrix \( U_V \ (\mu^*) \) that is only valid for a single point \( \mu^* \) but can be used for all reduced configuration variables \( q^*_r \in \mathbb{R}^n \) that may arise during the online evaluation, e.g., at each nonlinear iteration and time instance considered while numerically solving the equations of motion. Additionally, we set the reference configuration to equilibrium, which results in \( \hat{c} = 0 \) and guarantees solvability of the the two-term Taylor expression with \( m = n \).

### 3.2. Implementation and cost

Procedure 3 summarizes the offline/online strategy for implementing the RBS strategy for approximating the potential energy. This method satisfies the online computational

---

**Procedure 3: Reduced-basis sparsification for potential energy**

**Offline stage**

1. Determine the sampling matrix \( P \).

**Online stage (given \( \mu^* \))**

1. Compute \( V^T \nabla_{qq} V (q_0 (\mu^*) ; \mu^*) V \).
2. Compute \( P_1^T \nabla_{qq} V (q_0 (\mu^*) ; \mu^*) P_1 \).
3. Solve Equation (3.4) for \( U_V (\mu^*) \).
4. For any \( q^*_r \in \mathbb{R}^n \), set \( \hat{V}_r(q^*_r ; \mu^*) = V(q_0 (\mu^*) + P U_V (\mu^*) q^*_r ; \mu^*) \), and compute the gradient as \( \nabla_{q_r \hat{V}_r(q^*_r ; \mu^*)} = U_V (\mu^*)^T P^T \nabla_q V (q_0 (\mu^*) + P U_V (\mu^*) q^*_r ; \mu^*) \).

---

cost requirements of problem (II) with one exception: online step 1 incurs an \( N \)-dependent operation count. However, online steps 1–3 depend only on the online point \( \mu^* \) and not on the reduced configuration variables \( q^*_r \). Thus, these steps are performed only once per parameter instance, and their cost can be amortized over all online-queried values of \( q^*_r \). As a result, this does not preclude significant computational savings, as will be shown in the numerical results reported in Section. Note that online step 2 is equivalent to computing just \( O(n^2) \) entries of \( \nabla_{qq} V \), which can be completed at a cost independent of \( N \). Step 3 requires \( O(n^3) \) operations.

**Remark.** Most nonlinear reduced-order modeling methods assume ‘\( H \)-independence’, which states that the Jacobian of the vector-valued nonlinear function is sparse; in the present context, this corresponds to sparsity of the matrix \( \nabla_{qq} V \). When this assumption holds, the proposed methodology
incurs low online computational cost. This efficiency results from the fact that computing $P^T \nabla_q V(q_0 (\mu^*) + PU_V (\mu^*) q^*; \mu^*)$ in Step 4 of Procedure 3 requires that only $m$ components of the gradient $\nabla_q V$ be evaluated; if $H$-independence holds, then these $m$ components depend on only $O(m)$ components of the argument $q_0 (\mu^*) + PXq_\epsilon$, leading to an $N$-independent operation count.

Unfortunately, $H$-independence does not hold for some problems in Lagrangian dynamics. For example molecular-dynamics models can be characterized by a potential that includes interaction terms between all particles, resulting in a dense matrix $\nabla_{qq} V$. Here, the proposed method can still achieve efficiency by ‘centering’ the configuration space at equilibrium such that $q_0 (\mu) \equiv 0$, $\forall \mu \in \mathcal{D}$. In this case, the method requires computing only $m$ components of the argument $q_0 (\mu^*) + PXq_\epsilon$ in Step 4 of Procedure 3 regardless of the sparsity of the matrix $\nabla_{qq} V$. This efficiency is achievable due to the fact that the method injects ‘sparsification’ in the argument of the nonlinear function. This ability to achieve an $N$-independent operation count when $H$-independence is violated distinguishes this method from others in the literature.

4. Lagrangian dynamics formulation. We have now developed techniques to approximate parameterized reduced symmetric-positive-definite matrices and potential functions. In this section, we show how these methods enable us to achieve the objective of this work: preserving Lagrangian structure in model reduction for nonlinear mechanical systems. We begin by presenting the Lagrangian–dynamics formulation for such systems and highlighting critical problem structure. Later, Section 5 describes existing nonlinear model-reduction techniques and explains how they destroy structure in this context. Section 6 presents $H$-independence is violated distinguishes this method from others in the literature.

We consider parameterized, nonlinear simple mechanical systems, with a particular focus on structural-dynamics models constructed by a finite-element formulation. Such models are defined by a triple $(Q, g, V)$ parameterized by system parameters $\mu \in \mathcal{D}$. The parameters may describe variations in shape and material properties, for example. The triple is composed of:

- A differentiable configuration manifold $Q$. We take $Q = \mathbb{R}^N$ where $N$ denotes the number of degrees of freedom in the model, considered to be ‘large’ in this work.
- A parameterized Riemannian metric $g(v, w; \mu)$, where $v$ and $w$ belong to the tangent bundle of $Q$. We take $g(v, w; \mu) = v^T M(\mu) w$, where $M(\mu)$ denotes the $N \times N$ parameterized symmetric positive-definite mass matrix.
- A parameterized potential-energy function $V : Q \times \mathcal{D} \rightarrow \mathbb{R}$.

The kinetic energy of a simple mechanical system can be expressed as $T(q ; \mu) = \frac{1}{2} g(q, q ; \mu) = \frac{1}{2} q^T M(\mu) q$, where $q : [0, T] \rightarrow Q$ denotes the time-dependent configuration variables and $T$ denotes the final time. This leads to the following expression for the Lagrangian, which represents the difference between the kinetic and potential energies:

$$L(q, q ; \mu) = \frac{1}{2} g(q, q ; \mu) - V(q ; \mu) \quad (4.1)$$

$$= \frac{1}{2} q^T M(\mu) q - V(q ; \mu). \quad (4.2)$$

In many cases, the non-conservative force$^3$ consist of an applied external force and a dissipative force arising from Rayleigh viscous damping. This dissipative force derives from a positive-semidefinite dissipation function$^4$:

$$F(q ; \mu) = \frac{1}{2} \dot{q}^T C(\mu) \dot{q}, \quad (4.3)$$

where $C(\mu) = \alpha M(\mu) + \beta \nabla_{qq} V(q_0 (\mu^*) ; \mu)$. We consider non-conservative forces of the form $f(q, \dot{q}, t ; \mu) - \nabla_q F(q ; \mu)$, where $f$ denotes the external force that is derived from the Lagrange–D’Alembert variational principle.

---

$^3$Conservative forces can be handled by directly including them in the Lagrangian.

$^4$Non-viscously damped systems can also often be derived by a positive-semidefinite dissipation function [1].
Given the Lagrangian (4.2), one can derive the equations of motion for a simple mechanical system subject to an external force and Rayleigh viscous damping from the forced Euler–Lagrange equation

$$\frac{d}{dt} \nabla_{\dot{q}} L(q, \dot{q}; \mu) - \nabla_q L(q, \dot{q}; \mu) = f(q, \dot{q}, t; \mu) - \nabla_q F(q; \mu).$$

(4.4)

Substituting Eqs. (4.2) and (4.3) into Eq. (4.4) leads to the familiar equations of motion

$$M(\mu)\ddot{q} + C(\mu)\dot{q} + \nabla_q V(q; \mu) = f(q, \dot{q}, t; \mu).$$

(4.5)

Conservative mechanical systems, where \(f(q, \dot{q}, t; \mu) = 0\) and \(C(\mu) = 0\), exhibit important properties and can be characterized using the Hamiltonian formulation of classical mechanics discussed in Appendix A. For example, these systems conserve energy and quantities associated with symmetry, and their time-evolution maps are symplectic. Because these properties are intrinsic characteristics of the mechanical systems, it is desirable for numerical methods to preserve these properties. As mentioned in the introduction, the class of structure-preserving time integrators has been developed for this purpose. This class of integrators ensures that the numerical solution preserves essential properties such as energy conservation, momentum conservation, and symplecticity [15, 19].

For this reason, we aim to develop a reduced-order model that preserves the structure of the mechanical system, yet is computationally inexpensive to simulate. This will ensure that the reduced-order model preserves these characteristic properties. Further, the reduced-order equations of motion for these can be solved with a structure-preserving time integrator; this will ensure that the numerical solution computed using the reduced-order model will also preserve these properties. The properties of the system we seek to preserve are those enumerated in Section 1: a configuration space, a parameterized Riemannian metric, a parameterized potential-energy function, a parameterized positive-semidefinite dissipation function, and an external force derived from the Lagrange–D’Alembert principle. The first three properties constitute the parameterized triple that ensures the model describes a simple mechanical system; the last two characterize the non-conservative forces.

5. Existing model-reduction techniques. Model-reduction techniques aim to generate a low-dimensional model that is inexpensive to evaluate, yet captures the essential features of the high-fidelity model. These methods first conduct a computationally expensive offline stage during which they perform analyses (e.g., solving the equations of motion, modal analyses) for a training set \(\{\mu^i\}_{i=1}^p \subset D\). Then, these methods employ the data generated during these analyses to define a configuration manifold of reduced dimension, as well as other approximations to achieve efficiency in the presence of nonlinearities or arbitrary parameter dependence. This low-dimensional configuration manifold is subsequently employed to generate a low-dimensional model that can be used to perform inexpensive analyses for any specified point \(\mu^* \in D\) during the online stage.

When the configuration space is Euclidean (as is the case for the models considered herein), the configuration space of reduced dimension \(n \ll N\) can be expressed as

$$Q_r \equiv \{\bar{q}(\mu) + Vq_r \mid q_r \in Q_r\},$$

(5.1)

where \(\bar{q}(\mu) : D \to \mathbb{R}^N\) denotes the (parameterized) reference configuration about which the affine reduced subspace is centered, \(Q_r = \mathbb{R}^n\), and \(V \in \mathbb{R}^{N \times n}\) defines the reduced basis represented as a (typically dense) matrix. This leads to the following expression for the generalized coordinates and their derivatives:

$$q = \bar{q}(\mu) + Vq_r$$

(5.2)

$$\dot{q} = V\dot{q}_r$$

(5.3)

$$\ddot{q} = V\ddot{q}_r.$$  

(5.4)

Thus, the low-dimensional configuration space can be described in terms of low-dimensional generalized coordinates \(q_r \in Q_r\) or in terms of original coordinates by Eq. (5.2). The basis \(V\) can be determined by a variety of techniques, including proper orthogonal decomposition and modal decomposition.
5.1. Galerkin projection. Model reduction based on Galerkin projection preserves Lagrangian structure. As pointed out by Lall et al. [18], the Galerkin projection must be carried out on the Euler–Lagrange equation (4.5)—not the first-order state-space form—in order to preserve this structure.

Following their approach, Galerkin-projection-based methods replace the original configuration space $Q$, by the reduced-order configuration space $Q_r$ and subsequently derive the equations of motion in the usual way using a set of lower-dimensional generalized coordinates. In this way, the resulting model has an identical structure to the original problem.

For simple mechanical systems subject to non-conservative forces, this amounts to defining the Lagrangian as

$$L_r(q_r, \dot{q}_r; \mu) \equiv L(q(\mu) + Vq_r, V\dot{q}_r; \mu)$$

$$= \frac{1}{2} q_r^T V^T M(\mu) V q_r - V(q(\mu) + Vq_r; \mu) \tag{5.5}$$

and the dissipation function as

$$F_r(q_r; \mu) \equiv F(Vq_r; \mu)$$

$$= \frac{1}{2} q_r^T V^T C(\mu) V q_r. \tag{5.6}$$

The external force, which is derived based on the Lagrange–D’Alembert variational principle, is transformed by relation (5.2) into

$$f_r(q_r, \dot{q}_r, t; \mu) \equiv V^T f(q(\mu) + Vq_r, V\dot{q}_r, t; \mu). \tag{5.7}$$

Following Section 4, the forced Euler–Lagrange equation applied to the Lagrangian $L_r$, the dissipation function $F_r$, and the external force $f_r$ leads to the reduced-order equations of motion

$$\frac{d}{dt} \nabla_{q_r} L_r(q_r, \dot{q}_r; \mu) - \nabla_{q_r} L_r(q_r, \dot{q}_r; \mu) + \nabla_{q_r} F_r(q_r; \mu) = f_r(q_r, \dot{q}_r, t; \mu). \tag{5.8}$$

This can be rewritten as

$$V^T M(\mu) V \dot{q}_r + V^T C(\mu) V q_r + V^T \nabla_{q_r} V(q(\mu) + Vq_r; \mu) = V^T f(q(\mu) + Vq_r, V\dot{q}_r, t; \mu). \tag{5.9}$$

Note that Eq. (5.11) could have also been derived by applying Galerkin projection to the original Euler–Lagrange equation (4.5), i.e., making substitutions (5.2)–(5.4) and left multiplying the system of equations by $V^T$.

Thus, the Galerkin reduced-order model preserves the problem structure because it preserves all five Lagrangian properties:

I. a configuration space $Q_r = \mathbb{R}^n$, which relates to the original configuration space by Eq. (5.1).

II. a parameterized Riemannian metric $g_r(v_r, w_r; \mu) = v_r^T V^T M(\mu) V w_r$.

III. a parameterized potential-energy function $V_r(q_r; \mu) = V(q(\mu) + Vq_r; \mu)$.

IV. a parameterized positive-semidefinite dissipation function $F_r(q_r; \mu) = \frac{1}{2} q_r^T V^T C(\mu) V q_r$, and

V. an external force $f_r$ derived from applying the Lagrange–D’Alembert principle to the original external force $f$, but restricted to variations in the configuration space $Q_r$.

5.1.1. Computational bottleneck. Although the equations of motion (5.11) are low dimensional, they remain computationally expensive to solve when the operators exhibit arbitrary parameter dependence and the potential is nonlinear. The reason is simple: computing the low-dimensional components of (5.11) incurs large-scale operations due to the density of $V$. For example, the following steps are required to compute $V^T M(\mu^*) V$ for a specific $\mu^* \in D$ during the online stage:

i. Compute $M(\mu^*)$, which incurs $O(N\omega)$ flops, where $\omega$ denotes the average number of nonzeros per row of the matrix $M(\mu^*)$.

ii. Compute the product $M(\mu^*) V$, which incurs $O(N\omega n)$ flops.

iii. Compute the product $V^T (M(\mu^*) V)$, which incurs $O(Nn^2)$ flops.
Thus, the cost scales with the large dimension \( N \) of the original configuration manifold. The same analysis holds for the product \( \mathbf{V}^T \mathbf{C}(\mu^*) \mathbf{V} \).

If the potential energy \( \mathbf{V} \) exhibits a (general) nonlinear dependence on coordinates \( \mathbf{q} \), the situation worsens. In this case, the vector \( \nabla_{\mathbf{q}} \mathbf{V}(\mathbf{q}(\mu) + \mathbf{V}_\mathbf{q}; \mu^*) \) and product \( \mathbf{V}^T \nabla_{\mathbf{q}} \mathbf{V} \) must be computed for every instance of \( \mathbf{q} \). Similarly, \( \mathbf{V}^T \mathbf{f}(\mathbf{q}, \dot{\mathbf{q}}; t; \mu^*) \) must be computed for every time instance. Thus, a dimension reduction is generally insufficient to generate models with computational complexity independent of \( N \).

\textbf{Remark.} If the mass matrix is affine in functions of the parameters \( \mathbf{M}(\mu) = \sum \alpha_i(\mu) \mathbf{M}_i \) with \( \alpha_i : \mathcal{D} \to \mathbb{R} \) and \( \mathbf{M}_i \in \mathbb{R}^{N \times N} \), then products \( \mathbf{V}^T \mathbf{M} \mathbf{V} \) can be assembled offline, and \( \mathbf{V}^T \mathbf{M}(\mu^*) \mathbf{V} = \sum \alpha_i(\mu^*) [\mathbf{V}^T \mathbf{M}_i \mathbf{V}] \) can be computed in \( \mathcal{O}(n^2) \) floating-point operations (flops) during the online stage \cite{17, 21}. Similar low-complexity results can be obtained for the other terms if they can be similarly expressed in separable form. However, affine parameter dependence is a quite limiting scenario and does not generally hold.

### 5.2. Complexity reduction.

Several techniques have been developed to mitigate the computational bottleneck described in Section 5.1.1. Before applying projection, these methods compute (or sample) only a few entries of the vector-valued functions; other entries are not computed. In effect, this complexity-reduction strategy is equivalent to employing a sparse left-projection test basis. Such methods have been successfully applied to ODEs that do not exhibit particular structure. However, when applied to mechanical systems described by Lagrangian mechanics, these techniques destroy Lagrangian structure.

#### 5.2.1. Collocation.

Collocation approaches \cite{4, 23} compute only a subset of the full-order equations of motion (4.5) before applying Galerkin projection. That is, the reduced-order equations of motion (5.11) are approximated by

\[
\begin{align*}
\mathbf{V}^T \mathbf{P}^T \mathbf{M}(\mu) \mathbf{V}_{\mathbf{q}_r} &+ \mathbf{V}^T \mathbf{P}^T \mathbf{C}(\mu) \mathbf{V}_{\mathbf{q}_r} + \mathbf{V}^T \mathbf{P}^T \nabla_{\mathbf{q}} \mathbf{V}(\mathbf{q}(\mu) + \mathbf{V}_\mathbf{q}; \mu) \\
&= \mathbf{V}^T \mathbf{P}^T \mathbf{f}(\mathbf{q}(\mu) + \mathbf{V}_\mathbf{q}, \mathbf{V}_{\mathbf{q}_r}; t; \mu).
\end{align*}
\]

Recall that the sampling matrix \( \mathbf{P} \) consists of \( m \) selected columns of the identity matrix. If one considers the matrix \( \mathbf{P}^T \mathbf{V} \) as defining a basis for a test space, Eq. (5.12) can be viewed as a Petrov–Galerkin projection.

Computing the components of Eq. (5.12) is inexpensive in the case of \( H \)-independence, i.e., when the matrices \( \mathbf{M}, \mathbf{C}, \nabla_{\mathbf{q}} \mathbf{V}, \nabla_{\mathbf{q}} \mathbf{f}, \) and \( \nabla_{\mathbf{q}} \mathbf{f} \) are sparse. To see this, consider the first term in Eq. (5.12): computing \( (\mathbf{V}^T \mathbf{P})(\mathbf{P}^T \mathbf{M}(\mu^*)) \mathbf{V} \) for specific online point \( \mu^* \in \mathcal{D} \) incurs \( \mathcal{O}(m\omega n + mn^2) \) flops when operations are carried out in the order implied by the parentheses. This cost is small if the sparsity measure of \( \mathbf{M} \) is small, i.e., \( \omega \ll N \).

However, this cost-reduction approach destroys the problem’s structure, as it does not preserve the following Lagrangian properties described in Section 4.

\[ \boxed{\text{II}} \] The approximated reduced mass matrix \( \mathbf{V}^T \mathbf{P}^T \mathbf{M}(\mu) \mathbf{V} \) is not symmetric, so it does not define a metric.

\[ \boxed{\text{III}} \] The term \( \mathbf{V}^T \mathbf{P}^T \nabla_{\mathbf{q}} \mathbf{V}(\mathbf{q}(\mu) + \mathbf{V}_\mathbf{q}; \mu) \mathbf{V} \) is not symmetric, so it cannot be the Hessian of a potential-energy function.

\[ \boxed{\text{IV}} \] The approximated reduced damping matrix \( \mathbf{V}^T \mathbf{P}^T \mathbf{C}(\mu) \mathbf{V} \) is not symmetric, so it does not derive from a dissipation function.

Note that Property \[ \boxed{\text{II}} \] is trivially satisfied, as the configuration space can be described as \( \mathcal{Q}_r = \mathbb{R}^n \) and relates to the original configuration space by Eq. (5.1). Further, Property \[ \boxed{\text{IV}} \] is satisfied, because the non-conservative forces can be derived by applying the Lagrange–D’Alembert variational principle to a modified external force \( \mathbf{P}^T \mathbf{f}(\mathbf{q}, \dot{\mathbf{q}}; t; \mu) \), but restricted to variations in the (true) configuration space \( \mathcal{Q}_r \).

#### 5.2.2. DEIM/gappy POD.

Methods based on the discrete empirical interpolation method \cite{9, 13, 11} or gappy POD \cite{13, 6, 7} approximate via least-squares regression or interpolation the nonlinear vector-valued functions appearing in Eq. (4.5): these include \( \mathbf{M}(\mu) \mathbf{q}, \mathbf{C}(\mu) \mathbf{q}, \nabla_{\mathbf{q}} \mathbf{V}(\mathbf{q}; \mu) \), and \( \mathbf{f}(\mathbf{q}, \dot{\mathbf{q}}; t; \mu) \). Because these approaches construct a separate approximation for each term in the governing equations, they often achieve higher accuracy than collocation.

During the offline stage, these methods construct an orthogonal basis \( \mathbf{W}_\theta \in \mathbb{R}^{N \times n_\theta} \) with \( n_\theta \leq m \) for each nonlinear function \( \theta(t; \mu) \) appearing in the equations of motion. The basis \( \mathbf{W}_\theta \) can be computed empirically via proper orthogonal decomposition (POD), in which case the approximation technique is referred to as ‘gappy POD’ \cite{13}. This consists of two steps: 1) collect snapshots \( \mathcal{X}_\theta = \{ \theta(t; \mu) \mid t \in T_{\text{sample}}(\mu), \mu \in \{ \mu^1 \} \} \),
Here, we have used the notation
\[ \theta(t; \mu) \approx W_\theta [P^T W_\theta]^+ P^T \theta(t; \mu) \]  
where a superscript \( + \) denotes the Moore–Penrose pseudoinverse and \( [P^T W_\theta]^+ P^T \theta \) is simply the solution to the linear least-squares problem
\[ \min_{\theta \in \mathbb{R}^n} \| P^T \theta - P^T W_\theta \theta \|_2^2. \]  
Notice that when \( n_\theta = m \), the least-squares residual is zero (assuming the \( P^T W_\theta \) has full column rank) and so the above procedure corresponds to interpolation.

As with collocation, this approximation technique leads to computational-cost savings during the online stage if computing \( P^T \theta(t; \mu) \) incurs a flop count independent of \( N \), i.e., \( \theta(t; \mu) \) exhibits \( H \)-independence. Substituting least-squares approximations for the nonlinear functions into Eq. (5.11) yields the approximated reduced-order equations of motion
\[ Y_{M\theta} \dot{q} + Y_{C\theta} \ddot{q} + Y_{V\theta} \nabla_q V(q(\mu)) + W_{V\theta} \dot{q}_r = Y_t \bar{f}(q(\mu) + V_{q_r} \mu) = Y_t \bar{f}(q(\mu) + V_{q_r}, t; \mu). \]  
Here, we have used the notation
\[ Y_\theta \equiv V^T W_\theta [P^T W_\theta]^+ P^T, \]  
and the subscript of \( Y \) and \( W \) denotes the function for which the approximation has been constructed.

Unfortunately, this approximation method also destroys the Lagrangian structure. As before, Lagrangian properties \( \Box I \Box V \) are lost because the reduced mass, stiffness, and damping matrices are not symmetric. However, Property \( \Box I \) is preserved. Property \( \Box V \) is also preserved, because the non-conservative external force can be derived by the Lagrange–D’Alembert principle applied to the modified external force \( \bar{f}(q, q_r, t; \mu) = W_\theta [P^T W_\theta]^+ P^T \bar{f}(q, q_r, t; \mu) \) with variations restricted to the configuration space \( Q_r \).

6. Efficient, structure-preserving model reduction. The main idea of the proposed approach is to directly approximate the quantities defining the Lagrangian structure of the Galerkin-projection reduced-order model, and subsequently derive the equations of motion. Section 5 enumerates these quantities for the simple mechanical systems considered herein: the Riemannian metric \( g_r \), the potential-energy function \( V_r \), the semidefinite dissipation function \( \dot{F}_r \), and the external force \( \bar{f}_r \). Approximations to these ingredients should
1) preserve salient properties, 2) lead to computationally inexpensive reduced-order-model simulations, and 3) incur minimal approximation error.

To this end, we propose a model defined by
I. a configuration space \( Q_r = \mathbb{R}^n \), which relates to the original coordinates by Eq. (5.1),
II. an approximated Riemannian metric \( \bar{g}_r \),
III. an approximated potential-energy function \( \bar{V}_r \),
IV. an approximated positive-semidefinite dissipation function \( \bar{F}_r \), and
V. an approximated external force \( \bar{f}_r \) derived from applying the Lagrange–D’Alembert principle to an approximated force \( \bar{f}_r \) represented in the original coordinates, but limited to variations in the reduced configuration space \( Q_r \).

We can derive the equations of motion by applying the forced Euler–Lagrange equation with these approximations:
\[ \frac{d}{dt} \nabla_{q_r} \bar{L}_r(q, q_r; \mu) - \nabla_{q_r} \bar{L}_r(q, q_r; \mu) + \nabla_{q_r} \bar{F}_r(q, q_r; \mu) = \bar{f}_r(q, q_r, t; \mu), \]  
where the approximated Lagrangian is defined as
\[ \bar{L}_r(q, q_r; \mu) \equiv \frac{1}{2} \bar{g}_r(q, q_r; \mu) - \bar{V}_r(q, q_r; \mu). \]
Fig. 6.1. Comparing existing complexity-reduction approaches with the proposed approach. A dashed arrow implies a complexity-reduction approximation.

Note that Eq. (6.1) approximates Eq. (5.10), while Eq. (6.2) approximates Eq. (5.6).

Figure 6.1 depicts the strategy graphically. The next sections describe two proposed methods that align with this strategy for structure preservation. For reference, Table 6.1 reports components of the equations of motion for these methods, as well as for the model-reduction methods discussed in the previous sections.

### Table 6.1

| Method       | Mass matrix | damping matrix | potential-energy gradient | external force | struct. pres.? | low cost? |
|--------------|-------------|----------------|----------------------------|----------------|----------------|----------|
| Galerkin     | $V^T M(\mu)V$ | $V^T C(\mu)V$ | $V^T \nabla_q V(q_0(\mu) + V\xi_{q;\mu})$ | $V^T f$       | yes            | no       |
| collocation  | $V^T PP^T M(\mu)V$ | $Y_{Cq_c} C(\mu)V$ | $V^T PP^T \nabla_q V(q_0(\mu) + V\xi_{q;\mu}) + V^T PP^T f$ | $Y_t f$       | no             | yes      |
| gappy POD    | $Y_{Mr_c} M(\mu)V$ | $\alpha U_{Mr_c} M(\mu) U_{Mr} + \beta U_{Dr_c} \nabla_q V(q_0(\mu) + \xi_{q;\mu}) \mu V$ | $U_{Mr}^T \nabla_q V(q_0(\mu) + U_{Dr_c} \xi_{q;\mu})$ | $Y_t f$       | yes            | yes      |
| proposal 1   | $U_{Mr_c}^T M(\mu) U_{Dr_c}$ | $\alpha \sum_{i=1}^n \xi_{Mr_i} M(\mu) V + \beta U_{Dr_c} \nabla_q V(q_0(\mu) + \xi_{q;\mu}) \mu V$ | $U_{Mr}^T \nabla_q V(q_0(\mu) + U_{Dr_c} \xi_{q;\mu})$ | $Y_t f$       | yes            | yes      |

Terms appearing in the equations of motion for various model-reduction techniques, including the two proposed structure-preserving methods.

#### 6.1. Riemannian-metric approximation $\tilde{g}_r$. The function $g_r : (v_r; w_r; \mu) \mapsto v_r^T V^T M(\mu) V w_r$ is defined by a low-dimensional symmetric positive-definite matrix $V^T M(\mu) V$. Thus, the task of approximating this matrix is consistent with problem (1) of Section 2. We therefore propose computing an approximated Riemannian metric $\tilde{g}_r : \mathbb{R}^n \times \mathbb{R}^n \times \mathcal{D} \to \mathbb{R}$ as

$$\tilde{g}_r(v_1, v_2; \mu) \equiv v_1^T \tilde{M}(\mu) v_2,$$

where $\tilde{M}(\mu)$ is an $n \times n$ matrix that must be symmetric and positive definite. The first method (proposal 1 in Table 6.1) employs the reduced-basis sparsification technique, i.e., it approximates $\tilde{M}(\mu)$ via Eq. (5.21). The second method (proposal 2 in Table 6.1) employs matrix gappy POD and approximates this matrix by Eq. (5.26). Procedures 1 (Section 2.1) and 2 (Section 2.2) provide the offline and online steps to implement these approximations.
6.2. Potential-energy-function approximation $\tilde{V}_r$. Noting that only $\nabla_q \tilde{V}_r$ appears in the reduced-order equations of motion (see Eqs. (6.1)–(6.2)), problem (P2) of Section 3 applies to this scenario, and so we approximate the potential energy according to the method described in that section. In particular, Eq. (3.1) defines the approximated reduced potential energy. Further, we set the reference configuration to equilibrium $\tilde{q} = q_0$ to avoid the limitations associated with other choices (see the discussion in Section 3). Procedure 3 of Section 3.2 describes the offline/online decomposition for implementing this approximation.

6.3. Dissipation-function approximation $\tilde{f}_r$. To maintain the Rayleigh-damping structure, we simply approximate the damping matrix as a linear combination of the approximated mass matrix and Hessian of the potential at equilibrium

$$\tilde{f}_r(q; \mu) = \frac{\alpha}{2} v^T M(\mu) v + \frac{\beta}{2} v^T \nabla_q \tilde{V}_r(0; \mu) v,$$

(6.4)

where $\alpha$ and $\beta$ are the Rayleigh damping coefficients defined in Section 4.

6.4. External-force approximation $\tilde{f}_r$. The following form of the approximated external force preserves structure, i.e., ensures it is derived from applying the Lagrange–D’Alembert principle to an approximated force $\tilde{f}$ limited to variations in the reduced configuration space $Q_r$:

$$\tilde{f}_r(q_r; q_r, t; \mu) \equiv V^T \tilde{f}(q, \dot{q}, t; \mu).$$

(6.5)

Thus, the task of generating this approximation can be reduced to computing $\tilde{f}(q, \dot{q}, t; \mu)$—an approximation to the vector-valued function $f(q, \dot{q}, t; \mu)$. That is, we assign no special mathematical properties to $f$ aside from the fact that it is a vector. One way to accomplish this is by the DEIM/gappy POD approach described in Section 5.2.2.

The error in this approximation can be bounded using a result derived from the error in the gappy POD approximation of $f$ (e.g., see [7, Appendix D]). We obtain

$$\|\tilde{f}_r - f_r\|_2 \leq \|\left( I - W_f \left[ P^T W_f \right]^+ P^T \right) f\|_2 \leq \|R^{-1}\|_2 \|\left( I - W_f W_f^T \right) f\|_2,$$

(6.6)

where $W_f$ is an orthogonal basis used to represent the external force, and $P^T W_f = QR$ is the thin QR matrix factorization. This result assumes that $P^T W_f$ has full rank. Thus, the accuracy of this approximation relies both on the sampling matrix $P^T$ and how close $f$ is to the range of $W_f$. To achieve accuracy, we compute $W_f$ via POD, which minimizes the average value of $\|\left( I - W_f W_f^T \right) f\|_2^2$ over the training data.

6.4.1. Exactness conditions. Exactness conditions are similar to those described in Section 2.2.3 for the matrix gappy POD approximation. In the general case where $m < N$, if $f = 0$, then the approximation is exact, i.e., $\tilde{f}_r = f_r$. If instead $f$ has at least one non-zero entry, then sufficient conditions for an exact approximation are 1) $f \in \text{range}(W_f)$ and 2) $P^T W_f$ has full column rank. The first of these conditions holds, for example, when $W_f$ is computed via POD, the POD basis is not truncated, $f$ is independent of $q$ and $q$, $\mu \in \{\mu^*\}$, and if a snapshot of the external force was collected at the considered time instance. The second of these can be enforced by the method for choosing $P$, which is beyond the scope of this paper. In the full-sampling case where $m = N$, condition 2 holds automatically, so we only require condition 1 in this case.

6.4.2. Implementation. Procedure 4 provides the offline and online steps for implementing the external-force approximation.

7. Numerical experiments. Although the Galerkin and proposed reduced-order models have a theoretical advantage over the gappy POD and collocation reduced-order models in terms of preserving Lagrangian structure, it is unclear if this translates to improved numerical results in practice. This section investigates this question by applying the model-reduction techniques to a practical problem: the clamped-free truss structure shown in Figure 7.1.

We set the material properties to those of aluminum, i.e., density $\rho = 2700$ kg/m$^3$ and elastic modulus $E = 62 \times 10^9$ Pa. The external force is composed of four components:

$$f(\mu, t) = \sum_{i=1}^{4} r_i(\mu, t)r_i,$$

(7.1)
Procedure 4: External-force approximation via gappy POD

**Offline stage**

1. Collect snapshots of the external force $X_f \equiv \{ f(q, \dot{q}, t; \mu) \mid \mu \in \{\mu_i\}, \ t \in T_{\text{sample}}(\mu) \}$

2. Compute a POD basis $W_f$ using Algorithm 1 with inputs $X_f$ and $\eta_f \in [0,1]$.

3. Determine the sampling matrix $P$.

4. Compute the low-dimensional matrix $Y_f = V^T W_f [P^T W_f]^+$. 

**Online stage** (given $\mu^*$)

1. Compute $m \ll N$ entries of the external force $P^T f(q, \dot{q}, t; \mu^*)$.

2. Compute the low-dimensional matrix–vector product $Y_f [P^T f(q, \dot{q}, t; \mu^*)]$.

---

Fig. 7.1. Clamped–free parameterized truss structure

where $r_i \in \mathbb{R}^N$, $i = 1, \ldots, 4$ correspond to unit loads uniformly distributed across designated nodes and $r_i : \mathcal{D} \times [0, T] \to \mathbb{R}$, $i = 1, \ldots, 4$ denote the component-force magnitudes. Figure 7.1 depicts the spatial distribution of the forces, which lead to vectors $r_i$, $i = 1, \ldots, 4$ through the finite-element formulation described below. The parameterized, time-dependent magnitudes of these forces are

$$r_i(\mu, t) = \begin{cases} \gamma_i(\mu) \sin(\lambda_i(\mu)(t - T/4)), & t \geq T/4 \\ 0, & \text{otherwise} \end{cases},$$

(7.2)

where $\gamma_i : \mathcal{D} \to \mathbb{R}$ and $\lambda_i : \mathcal{D} \to \mathbb{R}$, $i = 1, \ldots, 4$ denote the maximum force magnitudes and force frequencies, respectively. Similarly, the initial condition is composed of four components

$$q(0; \mu) = \sum_{i=1}^{4} s_i(\mu)s_i,$$

(7.3)

where $s_i$ is the steady-state displacement of the truss subjected to load $r_i \gamma_i(\bar{\mu})$ with $\bar{\mu} = (0, \ldots, 0)$ denoting the nominal point in parameter space. The equilibrium configuration is simply the undeformed truss represented by $q_0(\mu) = 0$; thus, the configuration space is centered at equilibrium.

The truss is parameterized by 16 parameters $\mu \equiv (\mu_1, \ldots, \mu_{16}) \in [-1,1]^{16}$ that affect the geometry, initial condition, and applied force as described in Table 7.1. The problem is discretized by the finite-element method. The model consists of sixteen three-dimensional bar elements per bay with three degrees of freedom per node; this results in 12 degrees of freedom per bay. We consider a problem with 250 bays, which leads to $N = 3 \times 10^3$ degrees of freedom in the full-order model. The bar elements model geometric nonlinearity, which results in a high-order nonlinearity in the potential-energy $V$. 

---

18
This discretization leads to a model that corresponds to a Lagrangian dynamical system, with configuration manifold \( Q = \mathbb{R}^N \), Riemannian metric \( g(v, w; \mu) = v^T M(\mu) w \), nonlinear potential-energy function \( V \) and dissipation function \( F(q; \mu) \equiv \frac{1}{2} q^T C(\mu) q \). Here, \( C(\mu) = \alpha M(\mu) + \beta \nabla_{qq} V(0; \mu) \) corresponds to Rayleigh damping. Here, \( \alpha \) and \( \beta \) are chosen such that the damping ratio is a specified value \( \varsigma \) for the uncoupled ODEs associated with the smallest two eigenvalues of the matrix pencil \( (M(\mu), \nabla_{qq} V(0; \mu)) \).

To numerically solve the Lagrangian equations of motion in the time interval \([0, T]\) with \( T = 25 \) seconds, we employ the implicit midpoint rule (a symplectic integrator). This ensures that the numerical solution will yield symplectic time-evolution maps in the conservative case. We employ a globalized Newton solver with a More–Thuente linesearch \([12]\) to solve the system of nonlinear algebraic equations arising at each time step. Convergence of the Newton iterations is declared when the residual norm reaches \( 10^{-6} \) of its value computed using a zero acceleration and the values of the displacement and velocity at the beginning of the timestep. The linear system arising at each Newton iteration is solved directly.

The experiments compare the performance of four reduced-order models: Galerkin projection (Section 5.1), collocation (Section 5.2.1), and gappy POD (Section 5.2.2), and the proposed structure-preserving methods. All reduced-order models (ROMs) employ the same POD reduced basis \( V \), which is computed by applying Algorithm 1 with snapshots of the configuration variables and an energy criterion \( \eta_q \in [0, 1] \) specified within each experiment. The POD bases \( W_\theta \) employed by the gappy POD approach (see Section 5.2.2) are generated in the same way. In all cases, snapshots are only collected for the first half of the time interval at the training points; as a result, the second half of the time interval can be considered predictive—even for the training set.

Reduced-order models with complexity reduction employ the same sampling matrix \( P \), which is generated using GNAT’s greedy sample-mesh algorithm \([7]\). These models are also implemented using the sample-mesh concept \([7]\). To solve optimization problems \([2.2]\), we use the Poblano toolbox for unconstrained optimization \([12]\). The initial guess for each of these problems is chosen as \( P^T P V \). In practice, we always found the constraints to be inactive at the unconstrained solution to \([2.12]\); therefore, this reduces to a linear least-squares problem that we solve directly.

To compare the performance of the reduced-order models, we will consider the response quantity of interest to be the \( y \)-displacement of the bottom-left node of the end face of the truss in Figure 7.1; we denote this (parameterized, time-dependent) quantity by \( y \in \mathbb{R} \). The reported errors will be a normalized 1-norm (in time) of the error in this quantity:

\[
\text{error} = \frac{\sum_{t \in T_{\text{sample}}(\mu^*)} |y_{\text{ROM}}(t; \mu^*) - y_{\text{HFM}}(t; \mu^*)|}{|T_{\text{sample}}(\mu^*)| \left( \max_{t \in T_{\text{sample}}(\mu^*)} y_{\text{HFM}}(t; \mu^*) - \min_{t \in T_{\text{sample}}(\mu^*)} y_{\text{HFM}}(t; \mu^*) \right)}.
\]  

(7.4)

Here, \( y_{\text{ROM}} \) denotes the response computed by a reduced-order model, \( y_{\text{HFM}} \) is the high-fidelity ‘truth’ response, and \( T_{\text{sample}}(\mu^*) \subset [0, T] \) denotes the time instances selected by the time integrator for online point \( \mu^* \). In addition to the error in Eq. (7.4), we will compare the speedup achieved by the reduced-order models, measured as the ratio of the reduced-order-model simulation time to the full-order-model simulation time. All computations are carried out in Matlab on a Mac Pro with 2 × 2.93 GHz 6-Core Intel Xeon processors and 64 GB of memory.

\(^5\)We employ this error measure because it is insensitive to shifts in the average value of the displacement, unlike other measures such as the average 1-norm.
7.1. Conservative case. We first consider the conservative case characterized by zero damping $\zeta = 0$ and no external forces $\mu_i = -2$ for $i = 9, \ldots, 16$. This scenario is particularly interesting, as the full-order model corresponds to a conservative Lagrangian dynamical system characterized by energy conservation and symplectic time-evolution maps, and our method can also be interpreted as preserving Hamiltonian structure (see Appendix A). Because we numerically solve the equations of motion using the implicit midpoint rule, which is a symplectic integrator, the numerical solution is also characterized by a sympletic time-evolution map. This will also hold for reduced-order models that preserve Lagrangian structure, i.e., the Galerkin reduced-order model and the two proposed techniques (see Table 6.1). Note also that the dynamics of undamped, unforced structures are typically quite stiff, which often leaves reduced-order models prone to instabilities. As a result, we are free to vary parameters $\mu_i$, $i = 1, \ldots, 8$, which affect only the geometry and initial condition. We set the nominal forces that affect the initial condition to $f_1 = f_2 = 2\text{kg} \times 9.81\text{m/s}^2$ and $f_3 = f_4 = 0.4\text{kg} \times 9.81\text{m/s}^2$.

We first perform a timestep-verification study for the nominal point $\bar{\mu}$ characterized by $\bar{\mu}_i = 0$, $i = 1, \ldots, 8$ to ensure we employ an appropriate timestep in the numerical experiments. Results are shown in Figure 7.2. A timestep size of $\Delta t = 0.008$ seconds yields an observed convergence rate in the time-averaged tip displacement of 1.98, which is close to the asymptotic rate of convergence of the implicit midpoint rule, and an approximated error in the time-averaged tip displacement using Richardson extrapolation of $5.16 \times 10^{-7}$. We can therefore declare this to be an appropriate timestep size for the numerical experiments. Further, we note that the average number of Newton iterations per timestep is 3.15, so the geometric nonlinearity in the potential-energy function is significant.

Fig. 7.2. Conservative case: Timestep verification study at the nominal point. A timestep of $\Delta t = 0.008$ seconds gives a convergence rate of 1.98 and approximate relative error of $5.16 \times 10^{-7}$. Note that the three responses in the left figure nearly overlap.

7.1.1. Fixed parameters. We now test the model-reduction techniques in a fixed-parameters scenario. That is, we employ the nominal point in the parameter space for both the training and online points: $\{\mu^i\} = \bar{\mu}$ and $\mu^\star = \bar{\mu}$. Recall that we only collect snapshots for the first half of the time interval, so the second half can be considered a predictive regime. Note that the two proposed structure-preserving methods are the same for this case: they both exactly approximate the reduced mass matrix when the parameters are fixed.

The POD reduced basis $V$ is generated using an energy criterion of $\eta_q = 1 - 10^{-5}$ in Algorithm 1 of Appendix B, this leads to a basis dimension of only $n = 11 \ll N$. The gappy POD-based reduced-order model employs an energy criterion of 1 (i.e., no truncation) for its reduced bases $W_\theta$ (see Section 5.2.2). Figures 7.3 and 7.4 report results for the reduced-order models as the number of sample indices varies.\footnote{In all response plots, a ‘flat line’ indicates that the nonlinear solver failed to converge after 500 Newton iterations at three different time steps.}

In all response plots, a ‘flat line’ indicates that the nonlinear solver failed to converge after 500 Newton iterations at three different time steps.
First, note that the Galerkin reduced-order model is accurate (relative error of 5.42%); however, it yields a speedup of only 1.69. This is to be expected, as it preserves Lagrangian structure, but has no complexity-reduction mechanism (see Section 5.1.1). In addition, the proposed reduced-order model—which also preserves structure, yet has a complexity-reduction mechanism—yields a stable and accurate response regardless of the number of sample points chosen. For example, 0.4% sampling yields a relative error of 7.3% and a speedup of 207.0. Sampling 2% of the indices yields an error of 0.71% and a speedup of 34.5, and sampling 5% of the indices leads to 0.48% error and a speedup of 15.7. Note that sampling beyond 5% does not improve the method’s accuracy; however, it degrades the speedup, as it requires computing more entries of the vector-valued functions.

The other complexity-reducing reduced-order models (gappy POD and collocation) are always unstable except for collocation in the full-sampling case, when it is equivalent to Galerkin. This clearly highlights the practical benefits of preserving structure in model reduction, as existing structure-destroying complexity-reduction methods failed, even in the relatively simple scenario of fixed parameter values.

7.1.2. Varying parameters. We now consider a fully predictive scenario with $\mu \not\in \{\mu^i\}$. We use $p = 6$ training points and determine $\{\mu^i\}$ using Latin hypercube sampling. The online points are subsequently chosen randomly in the parameter space. Figure 7.5 depicts the tip displacement for the training points. Note that the responses are significantly different from one another. The two proposed structure-preserving reduced-order models will now be different from one another, as the parameters are varying, which means that the parameterized mass matrix will be approximated differently by the two techniques (see Table 6.1).

The reduced-order models employ a POD reduced basis with a truncation energy criterion of $\eta_q = 1 - 10^{-6}$, which yields a basis dimension of $n = 147 \ll N$. Again, the gappy POD-based reduced-order model employs a truncation criterion of $\eta_q = 1$ for its reduced bases. Figure 7.6 reports the tip displacements generated by the reduced-order models for the three randomly chosen online points, and Figure 7.7 reports the speedup and errors achieved by the reduced-order models as a function of the number of sample indices.

Again, note that the Galerkin reduced-order model is stable and accurate, as it generates relative errors of 18.7%, 14.5%, and 9.16% at the three online points, respectively. However, it yields discouraging speedups of 0.81 (i.e., the simulation was slower than for the full-order model), 1.61, and 1.32 at these points. The proposed structure-preserving methods are always stable and quite accurate. They yield nearly the same performance, although method two (which employs the matrix gappy POD approximation) generates lower errors for online points with 4.9% sampling. From Figure 7.6 note that the high-frequency oscillations that characterize the proposed methods’ responses are smoothed out when the sampling percentage reaches 20%.

7.2. Varying parameters. We will show in Section 7.2 that introducing dissipation improves the performance of both collocation and gappy POD. Note that gappy POD was also unstable for other attempted energy criteria of $\eta_q = 1 - 10^{-9}$ and $\eta_q = 1 - 10^{-8}$.
In particular, proposed method 2 generates speedups of 15.9, 28.5, and 26.2 and relative errors of 11.6%, 13.0%, and 11.6% for 4.9% sampling. For 20% sampling, the method generates speedups of 4.84, 9.82, and 7.72, and relative errors of 1.51%, 5.83%, and 1.09%.

In this example, the gappy POD reduced-order model is unstable for all sampling percentages, and the collocation reduced-order model is only stable for 100% sampling (at which point it is mathematically equivalent to the Galerkin reduced-order model). This is not surprising, as these methods do not preserve problem structure, nor do they guarantee energy conservation. This poor performance can be attributed to the stiff dynamics that characterize the considered conservative Lagrangian dynamical system, which lead to instabilities for both reduced-order models.

This example strongly showcases the practical importance of preserving Lagrangian structure: the proposed structure-preserving reduced-order models are the only models that yield both fast and accurate results.

7.2. Non-conservative case. We now consider the non-conservative case in which the non-conservative dissipative and external forces are nonzero. That is, we set $\zeta = \sin(5^\circ)$ and all parameters $\mu_i$, $i = 1, \ldots, 16$ are free to vary. We again set the nominal forces to $\bar{f}_1 = \bar{f}_2 = 2\text{kg} \times 9.81\text{m/s}^2$ and $\bar{f}_3 = \bar{f}_4 = 0.4\text{kg} \times 9.81\text{m/s}^2$.

As before, we perform a timestep-verification study for the nominal point $\bar{\mu}$ characterized by $\bar{\mu}_i = 0$, $i =$
Fig. 7.6. Conservative, varying-parameters case: reduced-order model responses as a function of sampling percentage $m/N \times 100\%$ for three randomly chosen online points. Legend: full-order model (black), Galerkin ROM (dark blue), structure-preserving ROM method 1 (magenta), structure-preserving ROM method 2 (light blue), gappy POD ROM (red), collocation ROM (green), end of training time interval (black circle).

1, . . . , 16 to discover an appropriate timestep. A timestep size of $\Delta t = 0.1$ seconds leads to an approximated error using Richardson extrapolation of $1.07 \times 10^{-4}$. We can therefore declare this to be an appropriate timestep size for the numerical experiments. Further, we note that the average number of Newton iterations per timestep is 2.56, so the nonlinearity remains significant.

7.2.1. Fixed parameters. We again test the different methods in the fixed-parameters case where $\{\mu_i\} = \bar{\mu}$ and $\mu^* = \bar{\mu}$. As above, we only collect snapshots for the first half of the time interval, and the proposed structure-preserving methods yield the same results.

The POD reduced basis $V$ is generated using an energy criterion of $\eta_q = 1 - 10^{-5}$, which leads to a
Fig. 7.7. Conservative, varying-parameters case: reduced-order model performance as a function of sampling percentage \( m/N \times 100\% \).
basis dimension of $n = 6 \ll N$. The gappy POD-based reduced-order model employs an energy criterion of 1 for its reduced bases $W_\theta$. Figures 7.8 and 7.9 report results for the reduced-order models as the number of sample indices varies.

Again, the Galerkin reduced-order model is accurate, with a relative error of 1.57%, but produces a speedup of only 1.33. The proposed structure-preserving method is always stable as expected. Its performance is dependent upon the sampling percentage, with (arguably) the best performance achieved for 2% sampling (6.28% error and 36.5 speedup). For 0.2% sampling, the method produces 16.1% error and a speedup of 251; 20% sampling leads to 5.39% error and a speedup of 4.6.

The gappy POD reduced-order model is unstable for 0.2%, 2%, and 5% sampling, but stabilizes at 20%; compared to the conservative case, this stability likely results from less stiff dynamics due to the presence of damping. This yields its best performance of 1.53% error, but only a 4.1 speedup. The collocation reduced-order model is stable only for full sampling, when it is equivalent to Galerkin.

![Fig. 7.8. Non-conservative, fixed-parameters case: reduced-order model responses as a function of sampling percentage $m/N \times 100\%$. Legend: full-order model (black), Galerkin ROM (dark blue), structure-preserving ROM (light blue), gappy POD ROM (red), collocation ROM (green), end of training time interval (black circle).](image1)

![Fig. 7.9. Non-conservative, fixed-parameters case: reduced-order model performance as a function of sampling percentage $m/N \times 100\%$. Missing data points for gappy POD and collocation ROMs indicate unstable responses.](image2)

---

8A truncation criterion of 1 yielded the best performance for Gappy POD. For $\eta_\theta = 1 \times 10^{-9}$, Gappy POD was unstable for all sampling percentages. It was also unstable for all sampling percentages when it employed an energy criterion of $\eta_\theta = 1 \times 10^{-8}$.
7.2.2. Varying parameters. We now consider the parameter-varying case where $\mu^* \not\in \{\mu^i\}$. We again employ $p = 6$ training points and determine $\{\mu^i\}$ using Latin hypercube sampling. We choose the online points randomly in the parameter space. Figure 7.10(a) shows the tip displacement for the training points; clearly, the responses are significantly different from one another.

![Graph](image)

**Fig. 7.10.** Non-conservative, parameter-varying case: tip displacement for the training set $\{\mu^i\}$ for two sets of experiments.

Because we are in a fully predictive scenario, the two proposed structure-preserving reduced-order models again yield different results. All reduced-order models employ an energy criterion of $\eta_q = 1 - 10^{-5}$, which leads to a basis dimension of $n = 12$. We employ $\eta_\theta = 1$ for the gappy POD reduced-order model.

Figures 7.11 and 7.12 report the results for this predictive study at the online points. At all three points, Galerkin is accurate (relative errors of 9.8%, 7.5%, and 13.5%), but does not yield significant speedups (speedups of 1.2, 1.4, and 1.1). As is apparent from the plots, the two proposed structure-preserving methods yield nearly the same performance. At 0.4% sampling, method 1 produces relative errors of 11.0%, 2.82%, and 10.3% and speedups of 73.3, 96.3, and 82.3. At 2% sampling, method 1 yields relative errors of 10.9%, 4.38%, and 7.97% and speedups of 19.2, 21.6, and 16.8.

In this example, gappy POD does not stabilize until 40% sampling, at which point the speedup is less than 1. Thus, gappy POD does not yield performance improvement for this problem. Collocation stabilizes at 80% sampling, and also fails to generate any performance improvement.

7.3. Effect of nonlinearity. We now aim to characterize the dependence of problem nonlinearity on the proposed methods’ performances. Recall from Section 3 that the potential-energy approximation is computed by matching the gradient of the potential energy to first order about the equilibrium configuration $q_0 (\mu^*)$. In the presence of stronger nonlinearity, we expect the configuration to deviate further from equilibrium, which should degrade the accuracy of the approximation.

To numerically assess the effect of nonlinearity, we repeat the experiments from Section 7.2.2 using the same training and online points, but we increase the nominal forces by a factor of 2.5 to $f_1 = f_2 = 5 \text{kg} \times 9.81 \text{m/s}^2$ and $f_3 = f_4 = 1 \text{kg} \times 9.81 \text{m/s}^2$. We first perform a timestep-verification study for the nominal point $\bar{\mu}$. As expected, a smaller timestep size of $\Delta t = 0.025$ seconds is required, as it corresponds to an approximated error using Richardson extrapolation of $3.62 \times 10^{-4}$.

Figure 7.10(b) displays the tip displacement for the training points. Note that the responses are similar to those for the previous study (see Figure 7.10(a)), but have larger magnitudes and therefore imply a stronger geometric nonlinearity. The reduced-order models employ a POD reduced basis of dimension $n = 14$, which was obtained by an energy criterion of $\eta_q = 1 - 10^{-5}$; gappy POD uses $\eta_\theta = 1$ for its nonlinear-function bases.

Figures 7.13 and 7.14 report the reduced-order models’ performances for this problem. As in the previous case, Galerkin is accurate (relative errors of 8.3%, 3.0%, and 10.0% at the online points), but does not
generate significant speedups (1.67, 1.71, and 1.0). The proposed structure-preserving techniques again yield very similar results to each other; however, the errors are significantly larger than in the experiments from Section 7.2.2 characterized by a less severe nonlinearity. For 0.5% sampling, proposed method 1 yields relative errors of 21.3%, 11.1%, and 15.9% at the online points and speedups of 116.4, 160, and 98.9. Thus, increasing the nonlinearity in the problem does have a deleterious effect on the methods’ performances.

However, it is important to note that other complexity-reducing reduced-order models fail to generate significant performance improvement on this more highly nonlinear problem. In particular collocation is always unstable for a sampling percentage less than 60%, and gappy POD is always unstable when the
Fig. 7.12. *Non-conservative, parameter-varying case: reduced-order model performance as a function of sampling percentage m/N × 100%.*
percentage is less than 80%. As a result, the best speedup obtained by either of the methods is only 2.77 (collocation for 60% sampling for online point 2).

Fig. 7.13. Non-conservative, highly nonlinear parameter-varying case: reduced-order model responses as a function of sampling percentage \( m/N \times 100\% \). Legend: full-order model (black), Galerkin ROM (dark blue), structure-preserving ROM method 1 (magenta), structure-preserving ROM method 2 (light blue), gappy POD ROM (red), collocation ROM (green), end of training time interval (black circle).

8. Conclusions. This paper has presented an efficient structure-preserving model-reduction strategy applicable to simple mechanical systems. The methodology directly approximates the quantities that define the problem’s Lagrangian structure and subsequently derives the equations of motion, while ensuring low online computational cost. The method is distinct from typical model-reduction methods for nonlinear ODEs; these methods are typically based on collocation and DEIM/gappy POD techniques that approximate the equations of motion and destroy Lagrangian structure. At the core of the methodology are the reduced-
Fig. 7.14. Non-conservative, highly nonlinear parameter-varying case: reduced-order model performance as a function of sampling percentage $m/N \times 100\%$. 
basis sparsification (RBS) and matrix gappy POD techniques for approximating parameterized reduced matrices while preserving symmetry and positive definiteness; we also employed the former method to preserve potential-energy structure.

Numerical experiments on a geometrically nonlinear parameterized truss structure highlight the method’s benefits: preserving Lagrangian structure ensured the method always generated stable responses that were often very accurate. Other model-reduction techniques were often unstable; achieving stability usually required too many sample indices to lead to significant performance gains for those methods. The experiments also showed that both RBS and matrix gappy POD led to nearly the same performance across a range of experiments.

Future work includes devising a method to improve the method’s robustness in the presence of strong nonlinearity (e.g., by non-local approximation of the potential-energy function), applying the method to a truly large-scale problem, devising a technique-specific method for choosing the sample indices, and deriving error bounds and error estimates that rigorously assess the accuracy of the method’s predictions. Finally, the RBS and matrix gappy POD methods are relevant to a wider class of problems than model reduction for Lagrangian systems; future work will investigate to their applicability to other scenarios.

Appendix A. Hamiltonian dynamics. When the Hamiltonian formulation of classical mechanics is taken, the proposed structure-preserving reduced-order models also preserve problem structure. For simplicity, we consider conservative systems with no dissipation or applied external forces. The Hamiltonian ingredients for conservative simple mechanical systems are then the same as those for Lagrangian dynamical systems as described in Section 1:

- A differentiable configuration manifold \( Q \), which we set to \( Q = \mathbb{R}^N \).
- A parameterized Riemannian metric \( g(v, w; \mu) \), which we set to \( g(v, w; \mu) = v^T M(\mu) w \), where \( M(\mu) \) denotes the \( N \times N \) parameterized symmetric positive-definite mass matrix.
- A parameterized potential-energy function \( V: Q \times D \rightarrow \mathbb{R} \).

Again, the kinetic energy can be expressed as \( T(\dot{q}; \mu) = \frac{1}{2} g(\dot{q}, \dot{q}; \mu) = \frac{1}{2} \dot{q}^T M(\mu) \dot{q} \), and the Lagrangian becomes \( L(q, \dot{q}; \mu) = \frac{1}{2} \dot{q}^T M(\mu) \dot{q} - V(q; \mu) \).

The conjugate momenta \( p : [0, T] \times D \rightarrow \mathbb{R}^N \) can then be derived as

\[
\dot{p}(t; \mu) = \frac{\partial L}{\partial \dot{q}} (t; \mu) = M(\mu) \dot{q}(t; \mu).
\]

In terms of the conjugate momenta, the kinetic energy then becomes \( T_q(p; \mu) = \dot{q}^T M(\mu)^{-1} p \). By definition, the Hamiltonian \( H : \mathbb{R}^N \times \mathbb{R}^N \times D \rightarrow \mathbb{R} \) is the Legendre transformation of the Lagrangian function:

\[
H(p, q; \mu) = \dot{q}^T p - L(q, \dot{q}, t)
\]

\[
= \frac{1}{2} p^T M(\mu)^{-1} p + V(q; \mu).
\]

Equivalently, it is the total energy, or sum of the kinetic and potential energies for classical mechanical systems. The equations of motion can then be obtained by applying Hamilton’s equations of motion

\[
\dot{q} = \frac{\partial H}{\partial p} \quad (A.5)
\]

\[
\dot{p} = -\frac{\partial H}{\partial q} \quad (A.6)
\]

which are equivalent to

\[
\ddot{q} = M(\mu) q \quad (A.7)
\]

\[
M(\mu) \ddot{q} + \nabla_q V(q; \mu) = 0 \quad (A.8)
\]

from the definition of the Hamiltonian and conjugate momenta. Note that these are equivalent to the equations of motion for a simple mechanical system \((4.5)\) in the conservative case derived using the Lagrangian formalism.
A.1. Galerkin reduced-order model. To construct the structure-preserving Galerkin reduced-order model for the Hamiltonian formalism, we follow this same recipe, but with the reduced ingredients defined previously. First, we define the reduced configuration space \( Q_r \in \mathbb{R}^n \) with \( Q_r \equiv \{ \bar{q}(\mu) + Vq_r \mid q_r \in Q_r \} \), and subsequently the reduced Lagrangian from Eqs. (5.5)–(5.6):

\[
L_r(q_r, \dot{q}_r; \mu) \equiv L(\bar{q}(\mu) + Vq_r, \dot{q}_r; \mu) = \frac{1}{2} q_r^T V^T M(\mu) Vq_r - V(\bar{q}(\mu) + Vq_r; \mu). 
\]

(A.9)

(A.10)

Then, the reduced conjugate momenta \( p_r : [0, T] \times D \to \mathbb{R}^n \) can be computed as

\[
p_r(t; \mu) = \frac{\partial L_r}{\partial \dot{q}_r}(t; \mu) = V^T M(\mu) Vq_r(t; \mu), 
\]

(A.11)

(A.12)

and the reduced Hamiltonian is, by definition,

\[
H_r(p_r, q_r; \mu) = \dot{q}_r^T p_r - L(q_r, \dot{q}_r, t) = \frac{1}{2} p_r^T (V^T M(\mu) V)^{-1} p_r + V(\bar{q}(\mu) + Vq_r; \mu). 
\]

(A.13)

(A.14)

Applying Hamilton’s equations of motion then yields

\[
\dot{q}_r = \frac{\partial H_r}{\partial p_r}, 
\]

(A.15)

\[
\dot{p}_r = -\frac{\partial H_r}{\partial q_r}, 
\]

(A.16)

or equivalently

\[
\dot{q}_r = V^T M(\mu) Vq_r, 
\]

(A.17)

\[
V^T M(\mu) Vq_r + V^T \nabla_q V(\bar{q}(\mu) + Vq_r; \mu) = 0
\]

(A.18)

Again, these are equivalent to the Galerkin reduced-order equations of motion obtained from the Lagrangian perspective in the absence of non-conservative forces. Clearly, this formulation for a reduced-order model preserves Hamiltonian structure, as the equations of motion are consistent with the Hamiltonian formalism.

A.2. Proposed structure-preserving methods. The proposed methods preserve Hamiltonian structure. The derivation follows that of the Galerkin reduced-order model in the previous section, but with the reduced Lagrangian replaced by the approximated reduced Lagrangian from Eq. (A.10):

\[
\tilde{L}_r(q_r, \dot{q}_r; \mu) = \frac{1}{2} \tilde{g}_r(q_r, \dot{q}_r; \mu) - \tilde{V}_r(q_r; \mu) = \frac{1}{2} q_r^T \tilde{M}(\mu) q_r - \tilde{V}_r(q_r; \mu), 
\]

(A.19)

(A.20)

where \( \tilde{M}(\mu) \) is an \( n \times n \) symmetric positive definite matrix generated by one of the methods presented in Section 2 and \( \tilde{V}_r(q_r; \mu) \) is an approximated potential approximated according to the method outlined in Section 3.

The approximated reduced conjugate momenta are then

\[
\tilde{p}_r(t; \mu) = \frac{\partial \tilde{L}_r}{\partial \dot{q}_r}(t; \mu) = \tilde{M}(\mu) q_r(t; \mu), 
\]

(A.21)

(A.22)
and the approximated reduced Hamilton is
\[
\hat{H}_r(\tilde{p}_r, \tilde{q}_r; \mu) = \dot{\tilde{q}}_r^T \tilde{p}_r - L(\tilde{q}_r, \dot{\tilde{q}}_r, t)
\]
\[
= \frac{1}{2} \tilde{p}_r^T \tilde{M}(\mu)^{-1} \tilde{p}_r + \tilde{V}_r(\tilde{q}_r; \mu).
\]

(A.23)

(A.24)

Applying Hamilton’s equations yields the following equations of motion
\[
\dot{\tilde{q}}_r = \tilde{M}(\mu) \tilde{q}_r
\]
\[
\tilde{M}(\mu) \ddot{\tilde{q}}_r + \nabla_{\tilde{q}_r} \tilde{V}_r(\tilde{q}_r; \mu) = 0
\]

which are computationally inexpensive to solve, and also retain Hamiltonian structure, as they are consistent with the Hamiltonian formalism. Note that again these are equivalent to the equations of motion obtained with the proposed strategy in the Lagrangian case with non-conservative forces set to zero.

**Appendix B. Proper orthogonal decomposition.** Algorithm 1 describes the method for computing a proper orthogonal decomposition (POD) basis given a set of snapshots. The method amounts to computing the singular value decomposition of the snapshot matrix; the left singular vectors define the POD basis.

**Algorithm 1** Proper-orthogonal-decomposition basis computation (normalized snapshots)

**Input:** Set of snapshots \( X \equiv \{x_i\}_{i=1}^{n_x} \subset \mathbb{R}^N \), energy criterion \( \eta \in [0, 1] \)

**Output:** \( W(X, \eta) \)

1. Compute the thin singular value decomposition \( X = U \Sigma V^T \), where \( X \equiv \left[ \begin{array}{c} \frac{x_1}{\|x_1\|} \cdots \frac{x_{n_x}}{\|x_{n_x}\|} \end{array} \right] \).
2. Choose dimension of truncated basis \( n = n_e(\eta) \), where \( n_e(\eta) \equiv \min_{i \in V(\eta)} i \) \( V(\eta) \equiv \{n \in \{1, \ldots, n_x\} | \sum_{i=1}^{n_x} \sigma_i^2 / \sum_{j=1}^{n_x} \sigma_j^2 \geq \eta \} \), (B.1)
3. \( W(X, \eta) = [u_1 \cdots u_n] \), where \( U \equiv [u_1 \cdots u_{n_x}] \).

**Appendix C. Solving the matrix gappy POD optimization problem.** This approach reformulates the constraints of problem (2.11) in terms of eigenvalues of the reduced matrix. That is, problem (2.11) is reformulated as
\[
\begin{aligned}
\text{minimize} & \quad \|P^T A(\mu)P - \sum_{k=1}^{n_A} P^T A_k P x_k \|^2_F \\
\text{subject to} & \quad \tilde{\lambda}_j(x) \geq \epsilon > 0, \quad j = 1, \ldots, n.
\end{aligned}
\]

(C.1)

Here, \( \tilde{\lambda}_j(x) \), \( j = 1, \ldots, n \) are the eigenvalues of the low-dimensional matrix \( \sum_{i=1}^{n_A} V^T A_i V x_i \) and \( \epsilon \) denotes a numerical threshold for defining a full-rank matrix. This problem can be numerically solved, e.g., using a gradient-based algorithm.

The gradient of the quadratic objective function is obvious. The gradient of the constraint can be derived by assuming distinct eigenvalues:
\[
\frac{\partial \tilde{\lambda}_j}{\partial x_i} = \tilde{y}_j^T \frac{\partial}{\partial x_i} \left( \sum_{k=1}^{n_A} V^T A_k V x_k \right) \tilde{y}_j
\]
\[
= \tilde{y}_j^T \left( V^T A_k V \right) \tilde{y}_j.
\]

(C.2)

(C.3)
Here, \( \tilde{y}_j \) is the eigenvector associated with eigenvalue \( \tilde{\lambda}_j \). This indicates that computing the gradient \( \frac{\partial \lambda_j}{\partial x_i} \) is inexpensive and requires the following steps:

1. Compute the eigenvector \( \tilde{y}_j \in \mathbb{R}^n \) of the matrix \( \sum_{k=1}^{n} \mathbf{V}_k \mathbf{A}_k \mathbf{V}_k \).

2. Compute the low-dimensional matrix–vector product \( \mathbf{w} = (\mathbf{V}_k \mathbf{A}_k \mathbf{V}_k) \tilde{y}_j \).

3. Compute the low-dimensional vector–vector product \( \tilde{y}_j^T \mathbf{w} \).

We propose using the unconstrained solution to problem (C.1) as the initial guess. In practice, this solution is often feasible, so it is typically unnecessary to handle the constraints directly. In the rare cases where it is necessary to deal with multiple equal eigenvalues—or a number of nearby eigenvalues—the methods presented by Andrew and Tan [3] can be used to produce a numerically stable gradient of the constraint; this was not required in the numerical experiments reported in Section 7.

Appendix D. Proofs.

D.1. Proof of Theorem 2.1. The proof relies on a generalization of the well known Cauchy interlacing theorem. To prove this generalization, we use a theorem (Theorem 4.3.10) from Ref. [16] that we restate below.

**Theorem D.1.** Let two sequences of interlacing real numbers be given by \( (\lambda_i^{(r)})_{i=1}^{n} \) and \( (\lambda_i^{(s)})_{i=1}^{m} \) as described by inequality (2.5) when \( m = n + 1 \). Define \( \Lambda^{(r)} = \text{diag}(\lambda_i^{(r)}) \) and \( \Lambda^{(s)} = \text{diag}(\lambda_i^{(s)}) \). Then, there exists a real number \( \alpha \in \mathbb{R} \) and a vector \( \mathbf{y} \in \mathbb{R}^n \) such that \( \Lambda^{(s)} \) are the eigenvalues of the real symmetric matrix

\[
\mathbf{B}^{(r)} = \begin{bmatrix} \Lambda^{(r)} & \mathbf{y}^T \\ \mathbf{y} & \alpha \end{bmatrix}.
\]

The following corollary is a direct consequence of the above theorem.

**Corollary D.2.** Given \( \mathbf{B}^{(r)} \in \text{SPD}(m) \) and \( \mathbf{B}^{(s)} \in \text{SPD}(m-1) \), where \( \text{SPD}(k) \) denotes the set of \( k \times k \) symmetric positive-definite matrices, whose eigenvalues interlace, then

\[
\exists \mathbf{U}_m \text{ such that } \mathbf{U}_m^T \mathbf{B}^{(r)} \mathbf{U}_m = \mathbf{B}^{(s)} \quad \text{with} \quad \mathbf{U}_m^T \mathbf{U}_m = \mathbf{I}.
\]  

**(D.1)**

**Proof.** Using the above theorem, a matrix \( \mathbf{B}^{(r)} \in \text{SPD}(m) \) exists that shares the same eigenvalues with \( \mathbf{B}^{(s)} \). Let \( \mathbf{Q}^{(r)} \), \( \mathbf{Q}^{(s)} \), and \( \mathbf{Q}^{(r)} \) be the (square, orthogonal) matrices of eigenvectors for \( \mathbf{B}^{(r)} \), \( \mathbf{B}^{(s)} \), and \( \mathbf{B}^{(r)} \), respectively. Then,

\[
\left( \mathbf{Q}^{(r)} \right)^T \mathbf{B}^{(r)} \mathbf{Q}^{(r)} = \Lambda^{(s)} = \mathbf{Q}^{(s)} \mathbf{B}^{(s)} (\mathbf{Q}^{(s)})^T,
\]

which implies that

\[
\mathbf{B}^{(r)} = \mathbf{Q}^{(r)} \Lambda^{(r)} (\mathbf{Q}^{(r)})^T = \mathbf{Q}^{(r)} \mathbf{B}^{(s)} (\mathbf{Q}^{(s)})^T.
\]

From the definition of \( \mathbf{B}^{(r)} \) it also follows that

\[
[\mathbf{I} \ 0] \mathbf{B}^{(r)} [\mathbf{I} \ 0]^T = \Lambda^{(r)},
\]

where \( \mathbf{I} \) is the \( (m-1) \times (m-1) \) identity matrix and \( \mathbf{0} \) is the zero column vector of length \( m-1 \), and thus

\[
\mathbf{B}^{(r)} = \mathbf{Q}^{(r)} \Lambda^{(r)} (\mathbf{Q}^{(r)})^T = \mathbf{Q}^{(r)} [\mathbf{I} \ 0] \mathbf{B}^{(s)} (\mathbf{Q}^{(s)})^T [\mathbf{I} \ 0]^T (\mathbf{Q}^{(r)})^T.
\]  

**(D.2)**

Combining the above, we can write

\[
\mathbf{B}^{(r)} = \mathbf{Q}^{(r)} [\mathbf{I} \ 0] \mathbf{Q}^{(s)} (\mathbf{Q}^{(s)})^T \mathbf{B}^{(s)} (\mathbf{Q}^{(s)})^T [\mathbf{I} \ 0]^T (\mathbf{Q}^{(r)})^T
\]

and so (D.1) is satisfied taking \( \mathbf{U}_m = [\mathbf{Q}^{(r)} [\mathbf{I} \ 0] \mathbf{Q}^{(s)} (\mathbf{Q}^{(s)})^T]^T. \)

\[ \square \]
The generalization of the Cauchy interlacing theorem now follows.

**Theorem D.3.** Given two matrices $\hat{B}^{(s)} \in \text{SPD} (m)$ and $\hat{B}^{(r)} \in \text{SPD} (n)$ (with $m \geq n$), then

$$\exists \; U \; \text{such that} \; U^T \hat{B}^{(s)} U = \hat{B}^{(r)} \; \text{with} \; U^T U = I \tag{D.3}$$

if and only if the eigenvalues $\lambda_i^{(r)}$, $i = 1, \ldots, n$ interlace the eigenvalues $\lambda_i^{(s)}$, $i = 1, \ldots, m$ defined as

$$\hat{B}^{(r)} \lambda_i^{(r)} = \lambda_i^{(r)} \hat{B}^{(r)}$$

$$\hat{B}^{(s)} \lambda_i^{(s)} = \lambda_i^{(s)} \hat{B}^{(s)}$$

where $\lambda_i^{(s)}$ denotes the $i$th smallest eigenvalue of the intermediate matrix. Rewriting this we obtain the following intervals for the eigenvalues $\mu_i$:

$$\mu_i \geq \begin{cases} \max(\lambda_i^{(s)}, \lambda_i^{(r)}) & i \geq m - n \\ \lambda_i^{(s)} & i < m - n \end{cases}$$

and

$$\mu_i \leq \begin{cases} \min(\lambda_{i+1}^{(s)}, \lambda_i^{(r)}) & i \leq n \\ \lambda_i^{(s)} & i > n \end{cases}$$

Using the interlacing property for $\hat{B}^{(s)}$ and $\hat{B}^{(r)}$, one can verify that the intervals for the $\mu_i$ are nonempty. That is, $\lambda_i^{(s)} \geq \lambda_i^{(s)}$ and for those $i$ such that $\lambda_i^{(r)} = \lambda_i^{(r)}$ is defined, we have $\lambda_i^{(s)} \geq \lambda_i^{(r)}$, $\lambda_i^{(s)} \geq \lambda_i^{(r)}$, and $\lambda_i^{(r)} \geq \lambda_{i-n+1}^{(r)}$. Thus, there exists an orthogonal matrix $U_m$ such that the $(m-1) \times (m-1)$ matrix $U^T_m \hat{B}^{(s)} U_m$ has eigenvalues that are interlaced by those of $\hat{B}^{(r)}$. We repeat this procedure each time reducing the matrix dimension by one until the final reduction where we take the lower dimension matrix to be equal to $\hat{B}^{(r)}$. This implies that there exists a set of projection matrices such that $U^T \hat{B}^{(s)} U = \hat{B}^{(r)}$ where $U = U_m U_{m-1} \ldots U_{n+1}$.

Equipped with the generalized Cauchy interlacing theorem, we now prove the exactness condition for the $A(\mu)$ term which is restated in slightly simplified notation.

**Theorem D.4.** Let $A(\mu)$ have the form

$$A(\mu) = h_1 (\mu) \; A_1 + h_2 (\mu) \; A_2 \tag{D.6}$$

where $A_1 \in \text{SPD} (N)$, $A_2 \in \text{SPD} (N)$, and $h_1, h_2 : \mathcal{D} \to \mathbb{R}$.

Then,

$$\exists \; U_A \; \text{such that} \; U_A^T P^T A(\mu) P U_A = V^T A(\mu) V, \; \forall \mu \in \mathcal{D} \tag{D.7}$$
if and only if the eigenvalues of the general matrix pencil

\[ \mathbf{B}(r)\mathbf{x}_i^{(r)} = \lambda_i^{(r)}\mathbf{D}(r)\mathbf{x}_i^{(r)}, \quad i = 1, \ldots, n \]  

interlace the eigenvalues of

\[ \mathbf{B}(s)\mathbf{x}_i^{(s)} = \lambda_i^{(s)}\mathbf{D}(s)\mathbf{x}_i^{(s)}, \quad i = 1, \ldots, m \]  

where

\[
\mathbf{D}(r) = \begin{bmatrix} \mathbf{V}_r^T \mathbf{A}_r \mathbf{V}_r \end{bmatrix}, \quad \mathbf{D}(s) = \begin{bmatrix} \mathbf{P}_r \mathbf{A}_r \mathbf{P}_r \end{bmatrix}, \\
\mathbf{B}(r) = \begin{bmatrix} \mathbf{V}_r^T \mathbf{A}_r \mathbf{V}_r \end{bmatrix}, \quad \text{and} \quad \mathbf{B}(s) = \begin{bmatrix} \mathbf{P}_r \mathbf{A}_r \mathbf{P}_r \end{bmatrix}.
\]

The definition of interlacing is given by

\[ \lambda_i^{(s)} \leq \lambda_i^{(r)} \leq \lambda_{i+m-n}^{(s)} \quad \text{for} \quad i = 1, \ldots, n \]  

where the eigenvalues are indexed in order of increasing magnitude.

**Proof.** Clearly (D.7) can only hold for any \( \mu \in \mathcal{D} \) and any functions \( h_1(\mu) \) and \( h_2(\mu) \) if and only if

\[
\mathbf{U}_\mathcal{A}^T \mathbf{D}(s) \mathbf{U}_\mathcal{A} = \mathbf{D}(r) \quad \text{and} \quad \mathbf{U}_\mathcal{A}^T \mathbf{B}(s) \mathbf{U}_\mathcal{A} = \mathbf{B}(r).
\]

Using a carefully chosen linear transformation, it follows that proving the theorem is equivalent to proving the following:

\[ \exists \mathbf{U} \text{ such that } \mathbf{U}^T \hat{\mathbf{B}}(s) \mathbf{U} = \hat{\mathbf{B}}(r) \quad \text{with} \quad \mathbf{U}^T \mathbf{U} = \mathbf{I} \]  

if and only if the eigenvalues \( \lambda_i^{(r)} \) interlace the eigenvalues of \( \lambda_i^{(s)} \), where the eigenvalues previously defined in Eqs. (D.8)–(D.9) also satisfy

\[
\hat{\mathbf{B}}(r)\hat{\mathbf{x}}^{(r)} = \lambda^{(r)}\hat{\mathbf{x}}^{(r)} \\
\hat{\mathbf{B}}(s)\hat{\mathbf{x}}^{(s)} = \lambda^{(s)}\hat{\mathbf{x}}^{(s)}.
\]

The linear transformation relies on Cholesky factorizations given by \( \mathbf{D}(s) = \mathbf{L}(s)(\mathbf{L}(s))^T \) and \( \mathbf{D}(r) = \mathbf{L}(r)(\mathbf{L}(r))^T \). These factors lead to the following definitions

\[
\hat{\mathbf{B}}(s) = (\mathbf{L}(s))^{-1}\mathbf{B}(s)(\mathbf{L}(s))^{-T}, \quad \hat{\mathbf{x}}^{(s)} = (\mathbf{L}(s))^T\mathbf{x}^{(s)}, \\
\hat{\mathbf{B}}(r) = (\mathbf{L}(r))^{-1}\mathbf{B}(r)(\mathbf{L}(r))^{-T}, \quad \hat{\mathbf{x}}^{(r)} = (\mathbf{L}(r))^T\mathbf{x}^{(r)}, \quad \text{and} \\
\mathbf{U} = (\mathbf{L}(s))^T \mathbf{U}_\mathcal{A} (\mathbf{L}(r))^{-T},
\]

which can be used in Eqs. (D.8), (D.9) and (D.11) to obtain Eqs. (D.12) and (D.13). The proof is completed by recognizing that Eqs. (D.12) and (D.13) correspond to the generalized Cauchy interlacing theorem.

\[ \blacksquare \]

**D.2. Proof of Theorem 2.2.** Proof. If condition 2 holds, then the unconstrained solution to problem (2.11) is

\[ \xi_\mathcal{A}(\mu^*) = (\mathbf{P}^T \mathbf{W}_\mathcal{A})^+ \mathbf{P}^T v(\mathbf{A}(\mu^*)). \]

If condition 1 holds, then the vectorized matrix can be expressed as

\[ v(\mathbf{A}(\mu^*)) = \mathbf{W}_\mathcal{A} \mathbf{z}(\mu^*), \]

or equivalently

\[ \mathbf{A}(\mu^*) = \sum_{i=1}^{n_\mathcal{A}} z^i(\mu^*) \mathbf{A}_i, \]
where $z \equiv (z^1, \ldots, z^{n_A})$. Substituting Eq. (D.15) into Eq. (D.14) gives $\xi_A(\mu^*) = z(\mu^*)$ and so Eq. (2.6) yields

$$\tilde{A}(\mu^*) = \sum_{i=1}^{n_A} z^i(\mu^*) V^T A_i V.$$  

(D.17)

Comparing Eqs. (D.17) and (D.16) gives the exactness result: $\tilde{A}(\mu^*) = V^T A(\mu^*) V^T$.

**D.3. Proof of Lemma 1.**

Proof. The equation $\tilde{U}_V = \tilde{U}_V = I$ simply states that the $m$ columns of $\tilde{U}_V$ are orthogonal and so any orthogonal matrix $\tilde{U}_V$ satisfies the first part of (3.8). Thus, solvability amounts to finding an orthogonal matrix $\tilde{U}_V$ such that $\tilde{V}^T \tilde{c} = \tilde{U}_V^T P^T \tilde{c}$. For a solution to exist, however, it is obviously necessary that $||\tilde{V}^T \tilde{c}||_2^2 = ||\tilde{U}_V^T P^T \tilde{c}||_2^2$. If the vector $P^T \tilde{c}$ lies within the span of the basis defined by the columns of $\tilde{U}_V^T$, then $\tilde{U}_V^T P^T \tilde{c}$ preserves its 2-norm and so $||\tilde{U}_V^T P^T \tilde{c}||_2^2 = ||P^T \tilde{c}||_2^2$. That is, application of $\tilde{U}_V^T$ corresponds to a rotation of $P^T \tilde{c}$ about the origin and so length is preserved. If instead the vector $P^T \tilde{c}$ lies only partially within the span of the orthogonal basis, then $||\tilde{U}_V^T P^T \tilde{c}||_2 < ||P^T \tilde{c}||_2$. That is, application of $\tilde{U}_V^T$ corresponds to a rotation of the component of $P^T \tilde{c}$ lying within the span of the orthogonal basis. This implies that a necessary condition for a solution to (3.8) is that

$$||\tilde{V}^T \tilde{c}||_2 \leq ||P^T \tilde{c}||_2.$$  

(D.18)

Case 1: $m = n$

$P^T \tilde{c}$ must lie within the range of $\tilde{U}_V$ (as it is a full rank square matrix) and so it is necessary to have equality in (D.18) when $m = n$. One possible $\tilde{U}_V$ in this case is obtained by first defining a $Q_1 \in \mathbb{R}^{n \times n}$ and a $Q_2 \in \mathbb{R}^{n \times n}$ such that the first row of $Q_1$ is $c_1^T \tilde{V}$ with $c_1 = 1/||\tilde{V}||_2$. Likewise, the first row of $Q_2$ is taken as $c_1^T P; \; \text{note that} \; \alpha_1$ also normalizes this row because we assume $||\tilde{V}^T \tilde{c}||_2 = ||P^T \tilde{c}||_2$. All remaining rows are chosen so that both $Q_1$ and $Q_2$ are orthogonal matrices. This gives

$$Q_1 \tilde{V}^T \tilde{c} = ||\tilde{V}^T \tilde{c}||_2 e_1 = ||P^T \tilde{c}||_2 e_1 = Q_2 P^T \tilde{c},$$

where $e_1$ is the first canonical unit vector of length $n$ (first element is one and all other $n-1$ components are zero). A suitable $\tilde{U}_V$ that solves (3.8) is then given by $\tilde{U}_V^T = Q_1^T Q_2$. Thus, equality in (D.18) is also sufficient when $m = n$.

Case 2: $m > n$

The matrix $\tilde{U}_V$ is now rectangular. One possible $\tilde{U}_V$ is obtained by defining $Q_1$ as before while instead defining an $m \times m$ orthogonal matrix $Q_{\text{full}}$ with the first row again set to $c_2 \tilde{V}^T P$ with $c_2 = 1/||P^T \tilde{c}||_2$. This gives

$$Q_1 \tilde{V}^T \tilde{c} = ||\tilde{V}^T \tilde{c}||_2 e_1 \; \text{and} \; Q_{\text{full}} P^T \tilde{c} = ||P^T \tilde{c}||_2 e_1$$

where $e_1 \in \mathbb{R}^{m \times 1}$ is the first canonical unit vector of length $m$. If $||P^T \tilde{c}||_2 = ||\tilde{V}^T \tilde{c}||_2$, then a suitable $\tilde{V}$ solving (3.8) is given by taking $Q_2$ to be the first $n$ rows of $Q_{\text{full}}$ (as $Q_1 \tilde{V}^T \tilde{c} = Q_2 P^T \tilde{c}$) and taking $\tilde{U}_V^T = Q_1^T Q_2$. If $||P^T \tilde{c}||_2 > ||\tilde{V}^T \tilde{c}||_2$, then we define a vector $y$ as an arbitrary linear combination of the last $m - n$ rows of $Q_{\text{full}}$ such that $y$ has unit norm. The first row of $Q_2$ is then taken as

$$\begin{align*}
(Q_2)_1 &= \alpha_3 \frac{P^T \tilde{c}}{||P^T \tilde{c}||_2} + \sqrt{1 - \alpha_3^2} y, \\
\end{align*}$$

where $(Q)_k$ denotes the $k$th row of a matrix $Q$ and $\alpha_3 = ||\tilde{V}^T \tilde{c}||_2/||P^T \tilde{c}||_2$. The remaining rows of $Q_2$ are simply $(Q_{\text{full}})_k$ for $k = 2, \ldots, n$. It is easy to verify that $Q_2$ is again orthogonal and that $Q_2 P^T \tilde{c} = ||\tilde{V}^T \tilde{c}||_2 e_1.$
Thus, $Q_2P^T c = ||V^T c||_2 e_1$ and $\tilde{U}_V = Q_2^T Q_2$ is a possible solution implying that $||P^T c||_2 \geq ||V^T c||_2$ is a necessary and sufficient condition when $m > n$. □

**Appendix E.** The authors acknowledge Julien Cortial for both insightful discussions and for providing the original nonlinear-truss code that was modified to generate the numerical results. The authors also acknowledge Clancey Rowley for useful comments received at the 2013 SIAM Conference on Computational Science and Engineering.

This research was supported in part by an appointment to the Sandia National Laboratories Truman Fellowship in National Security Science and Engineering, sponsored by Sandia Corporation (a wholly owned subsidiary of Lockheed Martin Corporation) as Operator of Sandia National Laboratories under its U.S. Department of Energy Contract No. DE-AC04-94AL85000. The authors also acknowledge support by the Department of Energy Office of Advanced Scientific Computing Research under contract 10-014804.

**REFERENCES**

[1] S. Adhikari, *Damping models for structural vibration*, PhD thesis, Cambridge University, September 2000.

[2] S.S. An, T. Kim, and D.L. James, Optimizing cubature for efficient integration of subspace deformations, ACM Transactions on Graphics (TOG), 27 (2008), p. 165.

[3] Alan L. Andrew and Roger C. E. Tan, Computation of derivatives of repeated eigenvalues and the corresponding eigenvectors of symmetric matrix pencils, SIAM J. Matrix Anal. Appl., 20 (1998), pp. 78–100.

[4] P. Astrid, S. Weiland, K. Willcox, and T. Backx, Missing point estimation in models described by proper orthogonal decomposition, IEEE Transactions on Automatic Control, 53 (2008), pp. 2237–2251.

[5] C. Beattie and S. Gugercin, Structure-preserving model reduction for nonlinear port-hamiltonian systems, in Decision and Control and European Control Conference (CDC-ECC), 2011 50th IEEE Conference on, IEEE, 2011, pp. 6564–6569.

[6] K. Carlberg, C. Bou-Mosleh, and C. Farhat, Efficient non-linear model reduction via a least-squares Petrov–Galerkin projection and compressive tensor approximations, International Journal for Numerical Methods in Engineering, 86 (2011), pp. 155–181.

[7] K. Carlberg, C. Farhat, J. Cortial, and D. Amsallem, The GNAT method for nonlinear model reduction: effective implementation and application to computational fluid dynamics and turbulent flows, Journal of Computational Physics, 242 (2013), pp. 623–647.

[8] K. Carlberg, R. Tuminaro, and P. Boggs, Efficient structure-preserving model reduction for nonlinear mechanical systems with application to structural dynamics, in AIAA Paper 2012-1969, 53rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference, Honolulu, Hawaii, April 23–26 2012.

[9] S. Chaturantabut and D. C. Sorensen, Nonlinear model reduction via discrete empirical interpolation, SIAM Journal on Scientific Computing, 32 (2010), pp. 2737–2764.

[10] I. Chowdhury and S.P. Dasgupta, Computation of Rayleigh damping coefficients for large systems, The Electronic Journal of Geotechnical Engineering, 8 (2003).

[11] M. Drohmann, B. Haasdonk, and M. Ohlberger, Reduced basis approximation for nonlinear parametrized evolution equations based on empirical operator interpolation, SIAM Journal on Scientific Computing, 34 (2012), pp. A937–A969.

[12] D.M. Dunlavy, T.G. Kolda, and E. Acar, Poblano v 1.0: A Matlab toolbox for gradient-based optimization, Sandia National Laboratories, Albuquerque, NM and Livermore, CA, Tech. Rep. SAND, 1222 (2010). See http://www.math.ucla.edu/~dunlavy/poblano.html

[13] R. Everson and L. Sirovich, Karhunen–Loève procedure for gappy data, Journal of the Optical Society of America A, 12 (1995), pp. 1657–1664.

[14] D. Galbally, K. Fidkowski, K. Willcox, and O. Ghavas, Non-linear model reduction for uncertainty quantification in large-scale inverse problems, International Journal for Numerical Methods in Engineering, 81 (2009), pp. 1581–1608.

[15] E. Hairer, C. Lubich, and G. Wanner, *Geometric numerical integration: structure-preserving algorithms for ordinary differential equations*, vol. 31, Springer Verlag, 2006.

[16] R.A. Horn and C.R. Johnson, *Matrix Analysis*, Cambridge University Press, 1990.

[17] K. Ito and SS Rasheed, A reduced basis method for control problems governed by PDEs, Control and estimation of distributed parameter systems, (1998), pp. 153–168.

[18] S. Lall, P. Krysl, and J.E. Marsden, Structure-preserving model reduction for mechanical systems, Physica D: Nonlinear Phenomena, 184 (2003), pp. 304–318.

[19] J.E. Marsden and M. West, *Discrete mechanics and variational integrators*, Acta Numerica, 10 (2001), p. 514.

[20] Beresford N Parlett, *The symmetric eigenvalue problem*, vol. 7, SIAM, 1980.

[21] C. Prud’homme, D. V. Rosas, K. Veroy, L. Machiels, Y. Maday, A. T. Patera, and G. Turinici, Reliable real-time solution of parameterized partial differential equations: Reduced-basis output bound methods, Journal of Fluids Engineering, 124 (2002), pp. 70–80.

[22] C. Prud’Homme, Dimitrios V Rovas, Karen Veroy, Luc Machiels, Yvon Maday, Anthony T Patera, Gabriel Turinici, et al., Reliable real-time solution of parameterized partial differential equations: Reduced-basis output bound methods, Journal of Fluids Engineering, 124 (2001), pp. 70–80.
[23] D. Ryckelynck, *A priori hyperreduction method: an adaptive approach*, Journal of Computational Physics, 202 (2005), pp. 346–366.

[24] T. Tonn, *Reduced-Basis Method (RBM) for Non-Affine Elliptic Parameterized PDEs*, PhD thesis, Ulm University, December 2011.

[25] B. Vandereycken, *Low-rank matrix completion by Riemannian optimization*, SIAM Journal on Optimization, 23 (2013), pp. 1214—1236.

[26] D. Wirtz, D. C. Sorensen, and B. Haasdonk, *A-posteriori error estimation for DEIM reduced nonlinear dynamical systems*, Preprint Series, Stuttgart Research Centre for Simulation Technology, (2012).