

SPACE-TIME LEAST-SQUARES PETROV-GALERKIN PROJECTION FOR NONLINEAR MODEL REDUCTION*

YOUNGSOO CHOI[†] AND KEVIN CARLBERG[‡]

Abstract. This work proposes a space-time least-squares Petrov–Galerkin (ST-LSPG) projection method for model reduction of nonlinear dynamical systems. In contrast to typical nonlinear model-reduction methods that first apply (Petrov–)Galerkin projection in the spatial dimension and subsequently apply time integration to numerically resolve the resulting low-dimensional dynamical system, the proposed method applies projection in space and time simultaneously. To accomplish this, the method first introduces a low-dimensional space-time trial subspace, which can be obtained by computing tensor decompositions of state-snapshot data. The method then computes discrete-optimal approximations in this space-time trial subspace by minimizing the residual arising after time discretization over all space and time in a weighted ℓ^2 -norm. This norm can be defined to enable complexity reduction (i.e., hyper-reduction) in time, which leads to space-time collocation and space-time Gauss–Newton with Approximated Tensors (GNAT) variants of the ST-LSPG method. Advantages of the approach relative to typical spatial-projection-based nonlinear model reduction methods such as Galerkin projection and least-squares Petrov–Galerkin projection include a reduction of both the spatial and temporal dimensions of the dynamical system, and a priori error bounds that bound the solution error by the best space-time approximation error and whose stability constants exhibit slower growth in time. Numerical examples performed on model problems in fluid dynamics demonstrate the ability of the method to generate orders-of-magnitude computational savings relative to spatial-projection-based reduced-order models without sacrificing accuracy for a fixed spatio-temporal discretization.

Key words. space-time projection, least-squares Petrov–Galerkin projection, residual minimization, model reduction, nonlinear dynamical systems

AMS subject classifications. 65L05, 65L06, 65L60, 65M15, 65M22, 68U20

DOI. 10.1137/17M1120531

1. Introduction. Reduced-order models (ROMs) of nonlinear dynamical systems are essential for enabling high-fidelity computational models to be used in many-query and real-time applications such as uncertainty quantification, design optimization, and control. Such ROMs reduce the *spatial dimensionality* of the dynamical system by performing a projection process on the governing system of nonlinear ordinary differential equations (ODEs). The resulting ROM is then resolved in time via numerical integration, typically with the same time integrator and time step employed for the high-fidelity model. Unfortunately, many applications require simulating the model over long time intervals, leading to high *temporal dimensionality* characterized by the

*Submitted to the journal's Methods and Algorithms for Scientific Computing section March 13, 2017; accepted for publication (in revised form) October 8, 2018; published electronically January 2, 2019.

<http://www.siam.org/journals/sisc/41-1/M112053.html>

Funding: This work was performed at Sandia National Laboratories and was supported by the LDRD program (project 190968). Sandia National Laboratories is a multimission laboratory managed and operated by National Technology and Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International, Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA-0003525. Lawrence Livermore National Laboratory is operated by Lawrence Livermore National Security, LLC, for the U.S. Department of Energy, National Nuclear Security Administration under contract DE-AC52-07NA27344.

[†]Work was performed while employed in the Extreme-scale Data Science and Analytics Department, Sandia National Laboratories, Livermore, CA 94550. Current address: Computational Engineering Division, Lawrence Livermore National Laboratory, Livermore, CA 94550 (choi15@llnl.gov).

[‡]Extreme-scale Data Science and Analytics Department, Sandia National Laboratories, Livermore, CA 94550 (ktcarlb@sandia.gov).

number of time instances in the time discretization. For example, many applications in fluid dynamics require long-time simulations to compute adequate statistics such as power spectral densities; structural-dynamics applications can demand long-time integration when structures undergo significant deformations; long-time integration is necessary to assess stability of planetary orbits [27]; molecular dynamics simulations [20] and condensed phase dynamics [42] also require long-time integration.

As such, ROMs are often characterized by low spatial dimensionality, but high temporal dimensionality, which can limit realizable computational savings in practice. It also renders ROMs ineffective in applications that demand a low temporal dimension for computational tractability. For example, a high temporal dimension can render intrusive uncertainty quantification methods (e.g., stochastic Galerkin [3, 25]) and simultaneous analysis and design (SAND) in PDE-constrained optimization [26, 36, 15, 16, 37, 39, 4] computationally intractable, as the dimension of the system of equations arising in such applications scales with the temporal dimension of the problem. Further, rigorous error bounds for these ROMs typically grow exponentially in time [38, 28, 34, 11], which renders certification challenging. This work aims to devise a model-reduction methodology that enables significant reduction in both the spatial and temporal dimensions of the dynamical system, while simultaneously producing error bounds that exhibit slower growth in time.

Several attempts have been made to address this temporal-complexity bottleneck in model reduction. First, several authors have demonstrated that larger stable time steps (and thus a smaller number of time instances) can be taken with a ROM relative to the high-fidelity model in the case of explicit time integration [29, 32, 43]. However, this approach is not always feasible, as many problems (e.g., compressible fluid dynamics, chemical kinetics) exhibit stiff dynamics that require implicit time integration, where the time step is limited by accuracy rather than stability. Increasing the time step in these contexts could significantly degrade time-discretization accuracy.

Time-parallel methods (e.g., parareal [31], PITA [23], and MGRIT [22]) aim to reduce the (serial) wall time incurred by a fine temporal discretization. These approaches enable dynamical-system simulations to be parallelized in the temporal domain, and are well suited for reduced-order models, as spatial parallelism alone quickly saturates for such low-dimensional models. However, while time-parallel methods can reduce the wall-time of such simulations, they do not reduce temporal dimensionality; in fact, they increase the total computational cost of simulations (as measured in core-hours).

More recently, a “forecasting” approach was proposed that employs time-domain data (i.e., snapshot-matrix right singular vectors) generated during the offline stage of model reduction to produce accurate forecasts of the solution during online ROM simulations via gappy proper orthogonal decomposition (POD) [21]. These forecasts can be used (1) to generate accurate initial guesses for the Newton solver at each time step [12], or (2) as an accurate coarse propagator to accelerate convergence of time-parallel methods [10]. While both approaches reduce the computational cost incurred by time integration (by reducing the total number of Newton iterations and the wall time, respectively), neither directly reduces the temporal dimension of the ROM.

Alternatively, space-time ROMs have been devised in the reduced basis [45, 46, 50, 49], POD-Galerkin [48, 6], and ODE-residual minimization [17] contexts. These approaches successfully reduce the temporal dimension of the underlying model by performing projection with a low-dimensional space-time basis. In addition, these methods can remove spurious temporal modes (e.g., unstable growth, artificial dissipation).

pation) from the state space, which can in principle lead to more accurate long-time responses. Further, space–time reduced-basis ROMs [45, 46, 50, 49] are equipped with error bounds that are observed to grow linearly (rather than exponentially) in the final time. While these approaches are quite promising, they exhibit several drawbacks in terms of applicability to general large-scale nonlinear dynamical systems. First, the space–time reduced-basis approaches require a space–time finite-element discretization for the high-fidelity model. Such discretizations are uncommon, as most computational models used in practice are constructed via spatial discretization (e.g., with a finite difference, finite volume, or finite element method) followed by time integration (e.g., with a linear multistep or Runge–Kutta scheme). Second, these space–time ROM approaches (with the exception of a collocation-like approach proposed in [17]) provide no mechanism for complexity reduction (i.e., hyper-reduction), which precludes these techniques from reducing the computational complexity in the presence of general nonlinearities. Further, the above approaches (with the exception of [6]) compute only a single space–time basis vector per training simulation. This can severely limit the dimensionality (and accuracy) of the resulting space–time ROM in the case of large-scale nonlinear dynamical-system models, where the number of training simulations may be limited by computational-cost considerations.

To this end, we propose a novel space–time least-squares Petrov–Galerkin (ST-LSPG) method that combines advantages of the above space–time ROM approaches, as it (1) reduces the spatial and temporal dimensions of the dynamical system; (2) is equipped with a priori error bounds that bound the solution error by the best space–time approximation error and whose stability constants exhibit subquadratic growth in time; (3) is applicable to general nonlinear dynamical-system models; (4) is equipped with hyper-reduction to reduce the complexity in the presence of general nonlinearities; and (5) can extract multiple space–time basis vectors from each training simulation via tensor decomposition. To realize these advantages, the approach adopts aspects of both the forecasting and space–time ROM approaches described above.

The original spatial-projection-based LSPG method [9, 11, 8] performed the following steps: (1) apply temporal discretization¹ to the system of ODEs characterizing the high-fidelity model, (2) introduce a low-dimensional *spatial* trial subspace, and (3) compute the solution in the spatial subspace that minimizes (in a weighted ℓ^2 -norm) the discrete residual arising at each time step. This approach does not reduce temporal dimensionality, as the ROM and high-fidelity model time steps are typically the same. Instead, the proposed ST-LSPG method executes the following steps: (1) apply time integration to the system of nonlinear ODEs with an implicit or explicit linear multistep method, (2) introduce a low-dimensional *space–time* trial subspace, and (3) compute the solution in the space–time trial subspace that minimizes the discrete residual over all space and time in a weighted ℓ^2 -norm. This norm can be selected to enable hyper-reduction based on both collocation and gappy POD; this is in analogy to collocation [30, 1, 41] and gappy POD [1, 11] methods applied to spatial-projection-based ROMs.

Specific contributions of this work include

- A novel ST-LSPG model-reduction method for parameterized nonlinear dynamical systems (section 4), including choices of weighting matrices to enable hyper-reduction (section 4.3).
- Several strategies for computing the “ingredients” characterizing the ST-LSPG method: the space–time trial subspace via tensor decomposition (sec-

¹When explicit time integration is used, LSPG projection is equivalent to Galerkin projection [8].

tion 5.1), the space-time residual basis in the case of space-time GNAT (section 5.2), the sampling matrix to enable space-time hyper-reduction (section 5.3), and the initial guess for the Gauss-Newton solver used to compute the ST-LSPG solution (section 5.4).

- A priori error bounds that enable the error in the ST-LSPG solution to be bounded by the best approximation error as measured in the ℓ^2 -norm over all time (Theorem 6.3), and the ℓ^∞ -norm over all time (Theorem 6.4). Critically, the stability constants for these bounds grow linearly and subquadratically in time, respectively (Remark 6.1).
- A posteriori error bounds that enable the error in the ST-LSPG solution to be bounded by the value of the objective function minimized by the method (Corollary 6.6).
- Numerical experiments that demonstrate the ability of the method to produce significant computational-cost savings relative to existing spatial-projection-based nonlinear ROMs without sacrificing accuracy (section 7).

Reference [17], which also proposed a space-time residual-minimizing projection applicable to parameterized nonlinear dynamical systems, is perhaps the most closely related work to the proposed technique. Our work can be distinguished from that contribution in several ways. First, our approach applies residual minimization to the discretized ODE (i.e., $O\Delta E$) rather than the time-continuous ODE over all time. This facilitates deriving error bounds with respect to the (fully discrete) full-order-model solution (section 6); [17] did not provide error bounds for the residual-minimizing approximation (it provides an error bound only for the best linear-subspace approximation). Also, [17] enforces the sum of generalized-coordinate values to equal one; our method does not require such a constraint, which can enable lower objective-function values. Further, our approach enables multiple space-time basis vectors to be extracted from each training simulation via tensor decomposition (section 4.3); [17] computes only a single space-time basis vector from each training simulation, which can severely limit the dimensionality of the space-time basis in practice. Further, our proposed approach provides several mechanisms for enabling hyper-reduction, i.e., complexity reduction of the low-dimensional model (section 4.3); [17] proposes one approach, which is analogous to the space-time collocation method described in section 4.3.2.

We proceed by describing the time-continuous and time-discrete representations of the full-order model in section 2, followed by a summary of the previously developed spatial-projection-based LSPG method in section 3. Then, we present the ST-LSPG method in section 4, followed by proposals for the ingredients characterizing the method in section 5. Section 6 provides error analysis, section 7 reports numerical experiments, and section 8 concludes the paper.

2. Full-order model. We begin by deriving time-continuous (ODE) and time-discrete ($O\Delta E$) formulations of the full-order model (FOM).

2.1. Time-continuous representation. We consider the FOM to be a parameterized nonlinear dynamical system characterized by a parameterized system of nonlinear ODEs

$$(2.1) \quad \frac{dx^*}{dt} = f(x^*, t; \mu), \quad x^*(0; \mu) = x^0(\mu),$$

where $t \in [0, T]$ denotes time with $T \in \mathbb{R}_+$ denoting the final time, and $x^*(t; \mu)$ denotes the time-dependent, parameterized state implicitly defined as the solution to

problem (2.1) with $\mathbf{x}^* : [0, T] \times \mathcal{D} \rightarrow \mathbb{R}^{N_s}$ and $\mathbf{x}^*(\cdot; \boldsymbol{\mu}) \in \mathbb{R}^{N_s} \otimes H$. Here, H denotes the set of sufficiently smooth functions from $[0, T]$ to \mathbb{R} (e.g., $H = H^1([0, T])$) under consideration and \otimes denotes the tensor product. Further, $\mathbf{f} : \mathbb{R}^{N_s} \times [0, T] \times \mathcal{D} \rightarrow \mathbb{R}^{N_s}$ with $(\mathbf{w}, \tau; \boldsymbol{\nu}) \mapsto \mathbf{f}(\mathbf{w}, \tau; \boldsymbol{\nu})$ denotes the velocity, which we assume to be nonlinear in at least its first argument, $\mathbf{x}^0 : \mathcal{D} \rightarrow \mathbb{R}^{N_s}$ denotes the initial state, and $\boldsymbol{\mu} \in \mathcal{D}$ denotes the parameters with parameter domain $\mathcal{D} \subseteq \mathbb{R}^{n_\mu}$.

2.2. Time-discrete representation: Linear multistep methods. We now introduce a (generally nonuniform) time discretization characterized by time step $\Delta t^n \in \mathbb{R}_+$ and time instances $t^n = t^{n-1} + \Delta t^n$, $n \in \mathbb{N}(N_t)$ with $t^0 = 0$, $N_t \in \mathbb{N}$, and $\mathbb{N}(N) := \{1, \dots, N\}$.² Applying a linear k -step method³ to numerically solve (2.1) using this discretization yields an ODE, which is characterized by the following system of nonlinear algebraic equations to be solved for the numerical solution $\mathbf{x}(t^n; \boldsymbol{\mu}) \in \mathbb{R}^{N_s}$ at each time instance:

$$(2.2) \quad \mathbf{r}^n(\mathbf{x}(t^n; \boldsymbol{\mu}), \dots, \mathbf{x}(t^{n-k(t^n)}; \boldsymbol{\mu}); \boldsymbol{\mu}) = \mathbf{0}, \quad n = 1, \dots, N_t.$$

Here, $k(t^n)(\leq n)$ denotes the number of steps used by the linear multistep method at time instance n and the residual is defined as

$$\begin{aligned} \mathbf{r}^n : (\mathbf{w}^0, \dots, \mathbf{w}^{k(t^n)}; \boldsymbol{\nu}) &\mapsto \sum_{j=0}^{k(t^n)} \alpha_j^n \mathbf{w}^j - \Delta t^n \sum_{j=0}^{k(t^n)} \beta_j^n \mathbf{f}(\mathbf{w}^j, t^{n-j}; \boldsymbol{\nu}) \\ &: \mathbb{R}^{N_s} \otimes \mathbb{R}^{k(t^n)+1} \times \mathcal{D} \rightarrow \mathbb{R}^{N_s}, \end{aligned}$$

where coefficients $\alpha_j^n, \beta_j^n \in \mathbb{R}$, $j = 0, \dots, k(t^n)$, define a particular linear multistep scheme, $\alpha_0^n \neq 0$, and $\sum_{j=0}^{k(t^n)} \alpha_j^n = 0$ is necessary for consistency. We note that the numerical solution satisfies

$$\mathbf{x}(\cdot; \boldsymbol{\mu}) \in \mathbb{R}^{N_s} \otimes \mathcal{H}$$

with \mathcal{H} the set of functions from $\{t^n\}_{n=0}^{N_t}$ to \mathbb{R} . We refer to $\mathbb{R}^{N_s} \otimes \mathcal{H}$ as the “FOM trial space” in which solutions are sought. An isomorphism exists between \mathcal{H} and \mathbb{R}^{N_t} provided by the (invertible) function \mathbf{g} that “unrolls” time according to the time discretization as

$$\begin{aligned} \mathbf{g} : \mathbf{u} &\mapsto [\mathbf{u}(t^1) \ \cdots \ \mathbf{u}(t^{N_t})] \\ &: \mathbb{R}^p \otimes \mathcal{H} \rightarrow \mathbb{R}^p \otimes \mathbb{R}^{N_t}, \end{aligned}$$

where $p \in \mathbb{N}$ is an arbitrary dimension. This yields an equivalent (discrete) representation of the FOM space as

$$(2.3) \quad \mathbf{g}(\mathbf{x}(\cdot; \boldsymbol{\mu})) = [\mathbf{x}(t^1; \boldsymbol{\mu}) \ \cdots \ \mathbf{x}(t^{N_t}; \boldsymbol{\mu})] \in \mathbb{R}^{N_s} \otimes \mathbb{R}^{N_t}.$$

3. Least-squares Petrov–Galerkin method. This section describes the trial subspace (section 3.1) and projection (section 3.2) employed by the original spatial-projection-based LSPG method [9, 11, 8].

²Note that by assuming a fixed time discretization, we do not allow for adaptive time-step selection; enabling the proposed ST-LSPG method to be applied in the context of adaptive time stepping is the subject of future research.

³We consider only linear multistep methods for simplicity of presentation. One could develop the proposed ST-LSPG method for alternative time integrators in a rather straightforward manner; see, e.g., [8], which develops LSPG models for explicit, fully implicit, and diagonally implicit Runge–Kutta schemes.

3.1. Spatial trial subspace. The original LSPG method applies spatial projection using a subspace $\mathcal{S} := \text{span}\{\phi_i\}_{i=1}^{n_s} \subseteq \mathbb{R}^{N_s}$ with $\dim(\mathcal{S}) = n_s \leq N_s$ (hopefully with $n_s \ll N_s$). Using this subspace, the LSPG method approximates the numerical solution at each time instance t^n , $n \in \mathbb{N}(N_t)$, as $\mathbf{x}(t^n; \boldsymbol{\mu}) \approx \tilde{\mathbf{x}}(t^n; \boldsymbol{\mu}) \in \mathbf{x}^0(\boldsymbol{\mu}) + \mathcal{S}$ or equivalently

$$(3.1) \quad \tilde{\mathbf{x}}(t^n; \boldsymbol{\mu}) = \mathbf{x}^0(\boldsymbol{\mu}) + \sum_{i=1}^{n_s} \phi_i \hat{x}_i(t^n; \boldsymbol{\mu}),$$

where $\hat{x}_i(\cdot; \boldsymbol{\mu}) \in \mathbb{R} \otimes \mathcal{H}$ with $\hat{x}_i(0; \boldsymbol{\mu}) = 0$, $i \in \mathbb{N}(n_s)$, denotes the generalized coordinates. This approach is equivalent to enforcing the approximated numerical solution $\tilde{\mathbf{x}}(t; \boldsymbol{\mu})$ with $\tilde{\mathbf{x}} : [0, T] \times \mathcal{D} \rightarrow \mathbb{R}^{N_s}$ to reside in an affine “spatial trial subspace”

$$(3.2) \quad \tilde{\mathbf{x}}(\cdot; \boldsymbol{\mu}) \in \mathbf{x}^0(\boldsymbol{\mu}) \otimes \mathcal{O} + \mathcal{S} \otimes \mathcal{H} \subseteq \mathbb{R}^{N_s} \otimes \mathcal{H},$$

where $\mathcal{O} \in \mathcal{H}$ is defined as $\mathcal{O} : \{t^n\}_{n=0}^{N_t} \rightarrow 1$. Noting that $\mathbf{1}_{N_t} = \mathbf{g}(\mathcal{O})$, where $\mathbf{1}_p$ denotes a p -vector of ones, we also have

$$(3.3) \quad \mathbf{g}(\tilde{\mathbf{x}}(\cdot; \boldsymbol{\mu})) \in \mathbf{x}^0(\boldsymbol{\mu}) \otimes \mathbf{1}_{N_t} + \mathcal{S} \otimes \mathbb{R}^{N_t} \subseteq \mathbb{R}^{N_s} \otimes \mathbb{R}^{N_t}.$$

Remark 3.1 (LSPG projection does not reduce the temporal dimension). Comparing (2.3) and (3.3) reveals that LSPG projection reduces spatiotemporal dimension of the problem from $\dim(\mathbb{R}^{N_s} \otimes \mathbb{R}^{N_t}) = N_s N_t$ to $\dim(\mathbf{x}^0(\boldsymbol{\mu}) \otimes \mathbf{1}_{N_t} + \mathcal{S} \otimes \mathbb{R}^{N_t}) = n_s N_t$. Thus, while the spatial dimension has been reduced from N_s to n_s , the temporal dimension N_t has not been reduced. This can preclude significant computational-cost savings when the original problem is characterized by a long time interval T , or if small time steps Δt^n are needed for accuracy (e.g., for stiff dynamical systems).

3.2. Spatial projection. The LSPG ROM computes an approximate solution by sequentially minimizing the discrete residual arising at each time instance, i.e.,

$$(3.4) \quad \tilde{\mathbf{x}}(t^n; \boldsymbol{\mu}) = \arg \min_{\mathbf{v} \in \mathbf{x}^0(\boldsymbol{\mu}) + \mathcal{S}} \left\| \mathbf{A} \mathbf{r}^n(\mathbf{v}, \tilde{\mathbf{x}}(t^{n-1}; \boldsymbol{\mu}), \dots, \tilde{\mathbf{x}}(t^{n-k(t^n)}; \boldsymbol{\mu}); \boldsymbol{\mu}) \right\|_2^2, \quad n = 1, \dots, N_t,$$

or equivalently

$$(3.5) \quad \hat{\mathbf{x}}(t^n; \boldsymbol{\mu}) = \arg \min_{\hat{\mathbf{v}} \in \mathbb{R}^{n_s}} \left\| \mathbf{A} \mathbf{r}^n(\mathbf{x}^0(\boldsymbol{\mu}) + \Phi \hat{\mathbf{v}}, \tilde{\mathbf{x}}(t^{n-1}; \boldsymbol{\mu}), \dots, \tilde{\mathbf{x}}(t^{n-k(t^n)}; \boldsymbol{\mu}); \boldsymbol{\mu}) \right\|_2^2, \quad n = 1, \dots, N_t,$$

where $\hat{\mathbf{x}}(\cdot; \boldsymbol{\mu}) \equiv [\hat{x}_1(\cdot; \boldsymbol{\mu}) \cdots \hat{x}_{n_s}(\cdot; \boldsymbol{\mu})]^T \in \mathbb{R}^{n_s} \otimes \mathcal{H}$. Here, $\mathbf{A} \in \mathbb{R}^{z \times N_s}$ is a weighting matrix, where $n_s \leq z (\leq N_s)$ is necessary for the residual Jacobian in the nonlinear least-squares problem (3.4)–(3.5) to be nonsingular. Examples of weighting matrices include $\mathbf{A} = \mathbf{I}_{N_s}$ in the case of unweighted LSPG, where \mathbf{I}_p denotes the $p \times p$ identity matrix; however, this choice precludes computational-cost savings, as all N_s elements of the residual vector \mathbf{r}^n (as well as its Jacobian) must be computed during each iteration when solving nonlinear least-squares problem (3.4)–(3.5). To reduce the computational complexity, a “hyper-reduction” approach is required that necessitates computing only a subset of residual elements. Particular weighting-matrix choices that lead to hyper-reduction include $\mathbf{A} = \mathbf{Z} \in \{0, 1\}^{n_z \times N_s}$, where $n_z \leq N_s$ and \mathbf{Z} comprises selected rows of the identity matrix \mathbf{I}_{N_s} in the case of collocation [30]; and $\mathbf{A} = (\mathbf{Z} \Phi_r)^+ \mathbf{Z} \in \mathbb{R}^{n_r \times N_s}$, where $\Phi_r \in \mathbb{V}_{n_r}(\mathbb{R}^{N_s})$ denotes a basis for the residual and a superscript $+$ denotes the Moore–Penrose pseudoinverse in the case of GNAT [11]. Here, $n_s \leq n_r \leq n_z (\leq N_s)$ is necessary for the residual Jacobian to be nonsingular, and $\mathbb{V}_k(\mathbb{R}^n)$ denotes the Stiefel manifold: the set of orthogonal k -frames in \mathbb{R}^n .

4. Space–time least-squares Petrov–Galerkin method. We now derive the proposed space–time least-squares Petrov–Galerkin (ST-LSPG) projection method. We begin by specifying the space–time trial subspace in section 4.1, followed by a description of the space–time least-squares Petrov–Galerkin projection process in section 4.2. Then, section 4.3 describes choices for the weighting matrix to enable hyper-reduction.

4.1. Space–time trial subspace. To reduce both the spatial and temporal dimensions of the full-order model, in analogy to (3.2) we enforce the approximated numerical solution $\tilde{\mathbf{y}}$ to reside in an affine “space–time trial subspace”

$$(4.1) \quad \tilde{\mathbf{y}}(\cdot; \boldsymbol{\mu}) \in \mathcal{ST} \subseteq \mathbb{R}^{N_s} \otimes \mathcal{H},$$

where $\mathcal{ST} := \mathbf{x}^0(\boldsymbol{\mu}) \otimes \mathcal{O} + \text{span}\{\boldsymbol{\pi}_i\}_{i=1}^{n_{st}} \subseteq \mathbb{R}^{N_s} \otimes \mathcal{H}$ with $\dim(\mathcal{ST}) = n_{st} \ll N_s N_t$, and we enforce $\boldsymbol{\pi}_i(0) = \mathbf{0}$, $i \in \mathbb{N}(n_{st})$, such that $\tilde{\mathbf{y}}(0; \boldsymbol{\mu}) = \mathbf{x}^0(\boldsymbol{\mu})$.

Thus, at a given time instance, the space–time LSPG method approximates the numerical solution at each time instance $n \in \mathbb{N}(N_t)$ as $\mathbf{x}(t^n; \boldsymbol{\mu}) \approx \tilde{\mathbf{y}}(t^n; \boldsymbol{\mu}) \in \mathcal{ST}$ or equivalently

$$(4.2) \quad \tilde{\mathbf{y}}(t^n; \boldsymbol{\mu}) = \mathbf{x}^0(\boldsymbol{\mu}) + \sum_{i=1}^{n_{st}} \boldsymbol{\pi}_i(t^n) \hat{y}_i(\boldsymbol{\mu}),$$

where $\hat{y}_i(\boldsymbol{\mu}) \in \mathbb{R}$, $i \in \mathbb{N}(n_{st})$, denotes the generalized coordinate of the ST-LSPG solution. Comparing (3.1) and (4.2) reveals that the space–time trial subspace enables time dependence of the approximated solution to be moved from the generalized coordinates to the basis vectors; this enables fewer generalized coordinates to be computed in order to characterize the complete space–time solution.

Introducing $\boldsymbol{\pi}_i := \mathbf{g}(\boldsymbol{\pi}_i) \in \mathbb{R}^{N_s} \otimes \mathbb{R}^{N_t}$ and $\mathcal{ST} := \mathbf{x}^0(\boldsymbol{\mu}) \otimes \mathbf{1}_{N_t} + \text{span}\{\boldsymbol{\pi}_i\}_{i=1}^{n_{st}}$, we can also write the space–time trial subspace in discrete form as

$$(4.3) \quad \mathbf{g}(\tilde{\mathbf{y}}(\cdot; \boldsymbol{\mu})) \in \mathcal{ST} \subseteq \mathbb{R}^{N_s} \otimes \mathbb{R}^{N_t}.$$

Remark 4.1 (space–time LSPG projection reduces the temporal dimension). Comparing (2.3) and (4.3) reveals that the proposed space–time LSPG trial subspace reduces the spatiotemporal dimension of the problem from $\dim(\mathbb{R}^{N_s} \otimes \mathbb{R}^{N_t}) = N_s N_t$ to $\dim(\mathcal{ST}) = n_{st}$. Because the spatiotemporal dimension n_{st} can be chosen to be independent of both the spatial and temporal dimensions N_s and N_t , respectively, the proposed method can reduce both the spatial and temporal dimensions of the FOM.

Remark 4.2 (space–time trial subspace can remove spurious temporal modes). Restricting the ST-LSPG state to lie in the space–time trial subspace $\mathbf{x}^0(\boldsymbol{\mu}) \otimes \mathcal{O} + \mathcal{ST}$ can enable spurious temporal modes (e.g., spurious time growth or dissipation) to be removed from the set of possible solutions. More precisely, if the subspace \mathcal{ST} is computed from training data as will be described in section 5.1, then this subspace will contain only temporal modes that have been observed during the training simulations.

4.2. Space-time least-squares Petrov-Galerkin projection. To derive the ST-LSPG projection, we begin by defining

$$\begin{aligned} \mathbf{r} : (\tau^n; \mathbf{w}; \boldsymbol{\nu}) &\mapsto \mathbf{r}^n(\mathbf{w}(\tau^n), \dots, \mathbf{w}(\tau^{n-k(\tau^n)}); \boldsymbol{\nu}) \\ &: \{t^n\}_{n=0}^{N_t} \times \mathbb{R}^{N_s} \otimes \mathcal{H} \times \mathcal{D} \rightarrow \mathbb{R}^{N_s}, \\ \mathbf{r} : (\tau^n; \hat{\mathbf{w}}; \boldsymbol{\nu}) &\mapsto \mathbf{r}\left(\tau^n; \mathbf{x}^0(\boldsymbol{\mu}) + \sum_{i=1}^{n_{st}} \boldsymbol{\pi}_i(\cdot) \hat{w}_i; \boldsymbol{\nu}\right) \\ &: \{t^n\}_{n=0}^{N_t} \times \mathbb{R}^{n_{st}} \times \mathcal{D} \rightarrow \mathbb{R}^{N_s}. \end{aligned}$$

Note that the space-time residuals $\mathbf{r}(\cdot; \mathbf{w}; \boldsymbol{\nu}), \mathbf{r}(\cdot; \hat{\mathbf{w}}; \boldsymbol{\nu}) \in \mathbb{R}^{N_s} \otimes \mathcal{H}$ are defined from the OΔE residual \mathbf{r}^n ; as such, they are defined directly from the integrator used to perform time discretization for the FOM. We now introduce a vectorization function

$$\begin{aligned} \mathbf{h} : \mathbf{u} &\mapsto \text{vec}(\mathbf{g}(\mathbf{u})) \\ &: \mathbb{R}^p \otimes \mathcal{H} \rightarrow \mathbb{R}^{pN_t} \end{aligned}$$

and define the vectorized residual as a function of the full state

$$\begin{aligned} \bar{\mathbf{r}} : (\mathbf{w}; \boldsymbol{\mu}) &\mapsto \mathbf{h}(\mathbf{r}(\cdot; \mathbf{w}; \boldsymbol{\mu})) \\ &: \mathbb{R}^{N_s} \otimes \mathcal{H} \times \mathcal{D} \rightarrow \mathbb{R}^{N_s N_t} \end{aligned}$$

and as a function of the generalized coordinates

$$\begin{aligned} \bar{\mathbf{r}} : (\hat{\mathbf{w}}; \boldsymbol{\nu}) &\mapsto \mathbf{h}(\mathbf{r}(\cdot; \hat{\mathbf{w}}; \boldsymbol{\nu})) \\ &: \mathbb{R}^{n_{st}} \times \mathcal{D} \rightarrow \mathbb{R}^{N_s N_t}. \end{aligned}$$

We define the inner product $(\mathbf{u}, \mathbf{v})_{\Theta} := \mathbf{h}(\mathbf{v})^T \Theta \mathbf{h}(\mathbf{u})$ and associated norm $\|\mathbf{u}\|_{\Theta} = \sqrt{(\mathbf{u}, \mathbf{u})_{\Theta}}$ for $\mathbf{u}, \mathbf{v} \in \mathbb{R}^p \otimes \mathcal{H}$ and $\Theta \in \text{SPSD}(pN_t)$, where $\text{SPSD}(p)$ denotes the set of $p \times p$ symmetric positive semidefinite matrices; we also define the inner product $(\mathbf{u}, \mathbf{v})_2 := \mathbf{h}(\mathbf{v})^T \mathbf{h}(\mathbf{u})$ and the associated norm $\|\mathbf{u}\|_2 = \sqrt{(\mathbf{u}, \mathbf{u})_2}$.⁴ Now, we propose computing the ST-LSPG solution by minimizing the residual in a weighted ℓ^2 -norm as

$$(4.4) \quad \tilde{\mathbf{y}}(\cdot; \boldsymbol{\mu}) = \arg \min_{\mathbf{v} \in \mathcal{S}^{\mathcal{T}}} \|\mathbf{r}(\cdot; \mathbf{v}; \boldsymbol{\mu})\|_{\bar{\mathbf{A}}^T \bar{\mathbf{A}}}^2 = \arg \min_{\mathbf{v} \in \mathcal{S}^{\mathcal{T}}} \|\bar{\mathbf{A}} \bar{\mathbf{r}}(\mathbf{v}; \boldsymbol{\mu})\|_2^2,$$

where $\bar{\mathbf{A}} \in \mathbb{R}^{\bar{z} \times N_s N_t}$ is a space-time weighting matrix and $n_{st} \leq \bar{z} (\leq N_s N_t)$ is necessary for the residual Jacobian in the nonlinear least-squares problem (4.4) to be nonsingular. We can also write problem (4.4) in terms of the generalized coordinates as

$$(4.5) \quad \hat{\mathbf{y}}(\boldsymbol{\mu}) = \arg \min_{\hat{\mathbf{v}} \in \mathbb{R}^{n_{st}}} \|\mathbf{r}(\cdot; \hat{\mathbf{v}}; \boldsymbol{\mu})\|_{\bar{\mathbf{A}}^T \bar{\mathbf{A}}}^2 = \arg \min_{\hat{\mathbf{v}} \in \mathbb{R}^{n_{st}}} \|\bar{\mathbf{A}} \bar{\mathbf{r}}(\hat{\mathbf{v}}; \boldsymbol{\mu})\|_2^2,$$

where $\hat{\mathbf{y}} \equiv [\hat{y}_1(\boldsymbol{\mu}) \cdots \hat{y}_{n_{st}}(\boldsymbol{\mu})]^T \in \mathbb{R}^{n_{st}}$ and (4.2) relates $\hat{\mathbf{y}}$ to $\tilde{\mathbf{y}}$.

Necessary first-order optimality conditions for problem (4.5) correspond to stationarity of the objective function; i.e., the solution $\hat{\mathbf{y}}(\boldsymbol{\mu})$ satisfies

$$(4.6) \quad \left(\mathbf{r}(\cdot; \hat{\mathbf{y}}(\boldsymbol{\mu}); \boldsymbol{\mu}), \frac{\partial \mathbf{r}}{\partial \hat{w}_i}(\cdot; \hat{\mathbf{y}}(\boldsymbol{\mu}); \boldsymbol{\mu}) \right)_{\bar{\mathbf{A}}^T \bar{\mathbf{A}}} = 0, \quad i \in \mathbb{N}(n_{st}),$$

⁴The metric will be rank deficient if we employ hyper-reduction as described in section 4.3.

where

$$\begin{aligned} \frac{\partial \mathbf{r}}{\partial \hat{w}_i}(t^n; \hat{\mathbf{y}}; \boldsymbol{\mu}) &= \frac{\partial \mathbf{r}}{\partial \mathbf{w}}\left(t^n; \mathbf{x}^0(\boldsymbol{\mu}) + \sum_{\ell=1}^{n_{st}} \boldsymbol{\pi}_{\ell}(\cdot) \hat{y}_{\ell}; \boldsymbol{\mu}\right) \boldsymbol{\pi}_i(\cdot) \\ &= \sum_{j=0}^{k(t^n)} \frac{\partial \mathbf{r}^n}{\partial \mathbf{w}^j}\left(\mathbf{x}^0(\boldsymbol{\mu}) + \sum_{\ell=1}^{n_{st}} \boldsymbol{\pi}_{\ell}(t^n) \hat{y}_{\ell}(\boldsymbol{\mu}), \dots, \mathbf{x}^0(\boldsymbol{\mu}) + \sum_{\ell=1}^{n_{st}} \boldsymbol{\pi}_{\ell}(t^{n-k(t^n)}) \hat{y}_{\ell}(\boldsymbol{\mu}); \boldsymbol{\mu}\right) \boldsymbol{\pi}_i(t^{n-j}) \\ &= \sum_{j=0}^{k(t^n)} \left[\alpha_j^n - \Delta t^n \beta_j^n \frac{\partial \mathbf{f}}{\partial \mathbf{w}}\left(\mathbf{x}^0(\boldsymbol{\mu}) + \sum_{\ell=1}^{n_{st}} \boldsymbol{\pi}_{\ell}(t^{n-j}) \hat{y}_{\ell}(\boldsymbol{\mu}), t^{n-j}; \boldsymbol{\mu}\right) \right] \boldsymbol{\pi}_i(t^{n-j}) \end{aligned}$$

denotes the elements of the test basis. We refer to this approach as a space–time least-squares Petrov–Galerkin (ST-LSPG) projection because (4.6) corresponds to a Petrov–Galerkin projection with test basis $\{\frac{\partial \mathbf{r}}{\partial \hat{w}_i}(\cdot; \hat{\mathbf{y}}; \boldsymbol{\mu})\}_{i \in \mathbb{N}(n_{st})}$ and also satisfies necessary conditions for the nonlinear least-squares problem (4.4)–(4.5).

As problem (4.4)–(4.5) is simply a nonlinear least-squares problem with \bar{z} equations in $n_{st} (\leq \bar{z})$ unknowns, we can solve it with the Gauss–Newton method, which leads to the following sequence of iterates for $k = 0, \dots, k_{\max}(\boldsymbol{\mu}) - 1$ given an initial guess $\hat{\mathbf{y}}^{(0)}$:

$$\begin{aligned} (4.7) \quad & \left(\frac{\partial \mathbf{r}}{\partial \hat{w}}(\cdot; \hat{\mathbf{y}}^{(k)}; \boldsymbol{\mu}) \delta \hat{\mathbf{y}}^{(k)}, \frac{\partial \mathbf{r}}{\partial \hat{w}}(\cdot; \hat{\mathbf{y}}^{(k)}; \boldsymbol{\mu}) \right)_{\bar{\mathbf{A}}^T \bar{\mathbf{A}}} = - \left(\mathbf{r}(\cdot; \hat{\mathbf{y}}^{(k)}; \boldsymbol{\mu}), \frac{\partial \mathbf{r}}{\partial \hat{w}}(\cdot; \hat{\mathbf{y}}^{(k)}; \boldsymbol{\mu}) \right)_{\bar{\mathbf{A}}^T \bar{\mathbf{A}}}, \\ (4.8) \quad & \hat{\mathbf{y}}^{(k+1)} = \hat{\mathbf{y}}^{(k)} + \alpha^{(k)} \delta \hat{\mathbf{y}}^{(k)}, \end{aligned}$$

where we set $\hat{\mathbf{y}}(\boldsymbol{\mu}) = \hat{\mathbf{y}}^{(k_{\max}(\boldsymbol{\mu}))}$ at convergence and $\alpha^{(k)} \in \mathbb{R}$ denotes a step length that can be computed to ensure global convergence (e.g., satisfy the strong Wolfe conditions). Again, we note that the condition $n_{st} \leq \bar{z}$ is necessary for singular values of the Jacobian $\bar{\mathbf{A}} \frac{\partial \mathbf{r}}{\partial \hat{w}}(\cdot; \hat{\mathbf{y}}^{(k)}; \boldsymbol{\mu}) \in \mathbb{R}^{\bar{z} \times n_{st}}$ to be uniformly bounded away from zero in the region of interest; this is one of the sufficient conditions required to prove convergence of the Gauss–Newton method (see, e.g., [35, Theorem 10.1]).

Remark 4.3 (simplification for block-diagonal weighting matrices). If the weighting matrix is block-diagonal, i.e., $\bar{\mathbf{A}} = \text{diag } \bar{\mathbf{A}}^n$ with $\bar{\mathbf{A}}^n \in \mathbb{R}^{\bar{z}^n \times N_s}$, $\bar{z}^n \leq N_s$, and $\bar{z} = \sum_{n=1}^{N_t} \bar{z}^n$, then problems (4.4) and (4.5) simplify to

$$(4.9) \quad \tilde{\mathbf{y}}(\cdot; \boldsymbol{\mu}) = \arg \min_{\mathbf{w} \in \mathcal{S}^{\mathcal{T}}} \sum_{n=1}^{N_t} \|\bar{\mathbf{A}}^n \mathbf{r}^n(\mathbf{w}; \boldsymbol{\mu})\|_2^2 \quad \text{and} \quad \hat{\mathbf{y}}(\boldsymbol{\mu}) = \arg \min_{\hat{\mathbf{v}} \in \mathbb{R}^{n_{st}}} \sum_{n=1}^{N_t} \|\bar{\mathbf{A}}^n \mathbf{r}(t^n; \hat{\mathbf{v}}; \boldsymbol{\mu})\|_2^2,$$

respectively.

Remark 4.4 (space–time Galerkin projection). Using the present formalism, we can also derive a (discrete) space–time Galerkin projection by enforcing Galerkin orthogonality rather than the Petrov–Galerkin orthogonality in (4.6). In particular, this space–time Galerkin method computes the solution $\hat{\mathbf{y}}_G(\boldsymbol{\mu})$ satisfying

$$(\mathbf{r}(\cdot; \hat{\mathbf{y}}_G; \boldsymbol{\mu}), \boldsymbol{\pi}_i(\cdot))_{\Theta} = 0, \quad i \in \mathbb{N}(n_{st}),$$

for some prescribed metric $\Theta \in \text{SPSD}(N_s N_t)$. However, because the Galerkin solution $\hat{\mathbf{y}}_G(\boldsymbol{\mu})$ does not associate with the solution to any optimization problem in general, we do not pursue this method further. We note that this approach is the discrete

counterpart to the continuous Galerkin projection proposed in [48, 6]; however, these contributions effectively employ $\Theta = \mathbf{I}_{N_s N_t}$ and thus provide no mechanism for hyper-reduction as we do in section 4.3.

4.3. Weighting matrix and hyper-reduction. Hyper-reduction refers to reducing the computational complexity of nonlinear ROMs by introducing approximations of the nonlinear functions. This is typically achieved via collocation (wherein the nonlinear functions are simply sampled) [1, 41, 30] or function-reconstruction approaches (e.g., gappy POD [21], empirical interpolation [5, 13]), wherein the nonlinear function is approximated from a sample of its entries via interpolation or least-squares regression [1, 7, 13, 24, 19, 9, 11]. In the case of (spatial-projection-based) LSPG, it has been shown that hyper-reduction can be realized by particular choices of the weighting matrix [8]. We now propose several choices for the space-time weighting matrix $\bar{\mathbf{A}}$, some of which will lead to hyper-reduction for the ST-LSPG method. All of these hyper-reduction methods will ensure that the computational cost of the ST-LSPG method is independent of both the spatial and temporal dimensions characterizing the FOM if the Jacobian of the space-time residual is sparse, i.e., the number of nonzeros in $\mathbf{h}(\frac{\partial r_i}{\partial \mathbf{w}}(t^n; \mathbf{v}; \boldsymbol{\mu})) \in \mathbb{R}^{N_s N_t}$, $i \in \mathbb{N}(N_s)$, $n \in \mathbb{N}(N_t)$, is “small” and is independent of both the spatial and temporal dimensions N_s and N_t , respectively. This is sometimes referred to as the H -independence condition [19] and is inherited directly from H -independence of the spatial residual \mathbf{r}^n , as $\text{nnz}(\mathbf{h}(\frac{\partial r_i}{\partial \mathbf{w}}(t^n; \mathbf{v}; \boldsymbol{\mu}))) = \sum_{j=0}^{k(t^n)} \text{nnz}(\frac{\partial r_i^n}{\partial \mathbf{w}^j}(\mathbf{v}^n, \dots, \mathbf{v}^{n-k(t^n)}; \boldsymbol{\mu}))$, where nnz denotes the number of nonzeros of its argument.

4.3.1. Unweighted LSPG. The most obvious choice for the weighting matrix is $\bar{\mathbf{A}} = \mathbf{I}_{N_s N_t}$; this choice simply minimizes the sum of squares of elements of the residual over both space and time and leads to $\bar{z} = N_s N_t$ with $\bar{\mathbf{A}}^n = \mathbf{I}_{N_s}$, $n \in \mathbb{N}(N_t)$, in problem (4.9). This is analogous to unweighted LSPG in the spatial-projection case, in which case $\mathbf{A} = \mathbf{I}_{N_s}$ in problem (3.4)–(3.5).

However, because this approach requires evaluating all $N_s N_t$ of the space-time residual in order to compute the objective function, it precludes significant computational-cost savings. As was also pointed out in [6], this bottleneck is especially cumbersome for space-time ROM approaches. For this reason, alternative choices for the weighting matrix $\bar{\mathbf{A}}$ can be employed that lead to a ROM whose computational complexity is independent of the full spatiotemporal dimension $N_s N_t$. We now describe different choices for the weighting matrix $\bar{\mathbf{A}}$ that lead to such hyper-reduction.

4.3.2. Space-time collocation. We can extend collocation hyper-reduction to the ST-LSPG context by employing weighting matrix $\bar{\mathbf{A}} = \bar{\mathbf{Z}}$ with

$$(4.10) \quad \bar{\mathbf{Z}} := [\mathbf{e}_{\varphi(\delta_1, t_1)} \ \cdots \ \mathbf{e}_{\varphi(\delta_{\bar{n}_z}, t_{\bar{n}_z})}]^T \in \{0, 1\}^{\bar{n}_z \times N_s N_t},$$

where $\varphi : (i, j) \mapsto i + N_s(j - 1)$, and \mathbf{e}_i denotes the i th canonical unit vector. Here, $\mathbf{st} := \{(\delta_i, t_i)\}_{i \in \mathbb{N}(\bar{n}_z)} \subseteq \mathbb{N}(N_s) \times \mathbb{N}(N_t)$ denotes the set of space-time sample indices. Again, we require $n_{st} \leq \bar{n}_z (\leq N_s N_t)$ to ensure nonsingular residual Jacobians.

Critically, note that applying $\bar{\mathbf{A}} = \bar{\mathbf{Z}}$ in problem (4.4)–(4.5) leads to hyper-reduction, as evaluating the objective function requires evaluating only $\bar{n}_z < N_s N_t$ elements of the spatiotemporal residual. In practice, this implies that the residual will be evaluated only at a subset of time instances and spatial degrees of freedom. Further, this can also lead to a positive semidefinite metric $\Theta = \bar{\mathbf{A}}^T \bar{\mathbf{A}}$, as $\text{rank}(\bar{\mathbf{A}}^T \bar{\mathbf{A}}) = \bar{n}_z \leq N_s N_t$ in this case.

4.3.3. Space–time GNAT. Similarly, we can extend gappy POD hyper-reduction to ST-LSPG by employing a weighting matrix $\bar{\mathbf{A}} = (\bar{\mathbf{Z}}\bar{\Phi}_r)^+\bar{\mathbf{Z}} \in \mathbb{R}^{\bar{n}_r \times N_s N_t}$, where $\bar{\Phi}_r \in \mathbb{V}_{\bar{n}_r}(\mathbb{R}^{N_s N_t})$ denotes a basis (in matrix form) for the spatiotemporal residual. Again, we require $n_{st} \leq \bar{n}_r \leq \bar{n}_z (\leq N_s N_t)$ for nonsingular residual Jacobians. When this weighting matrix is employed, problem (4.4)–(4.5) is equivalent to minimizing the ℓ^2 -norm of the gappy POD-approximated residual; i.e., problem (4.4)–(4.5) with $\bar{\mathbf{A}} = (\bar{\mathbf{Z}}\bar{\Phi}_r)^+\bar{\mathbf{Z}} \in \mathbb{R}^{\bar{n}_r \times N_s N_t}$ is equivalent to

$$\tilde{\mathbf{y}}(\cdot; \boldsymbol{\mu}) = \arg \min_{\mathbf{v} \in \mathcal{S}\mathcal{T}} \|\tilde{\mathbf{r}}(\cdot; \mathbf{v}; \boldsymbol{\mu})\|_2^2 \quad \text{and} \quad \hat{\mathbf{y}}(\boldsymbol{\mu}) = \arg \min_{\hat{\mathbf{v}} \in \mathbb{R}^{n_{st}}} \|\tilde{\mathbf{r}}(\cdot; \hat{\mathbf{v}}; \boldsymbol{\mu})\|_2^2,$$

where we have defined the gappy POD residual approximations as

$$\begin{aligned} \mathbf{h}(\tilde{\mathbf{r}}(\cdot; \mathbf{v}; \boldsymbol{\mu})) &= \arg \min_{\bar{\mathbf{v}} \in \text{range}(\bar{\Phi}_r)} \|\bar{\mathbf{Z}}\bar{\mathbf{v}} - \bar{\mathbf{Z}}\bar{\mathbf{r}}(\mathbf{v}; \boldsymbol{\mu})\|_2^2 = \bar{\Phi}_r(\bar{\mathbf{Z}}\bar{\Phi}_r)^+\bar{\mathbf{Z}}\bar{\mathbf{r}}(\mathbf{v}; \boldsymbol{\mu}), \\ \mathbf{h}(\tilde{\mathbf{r}}(\cdot; \hat{\mathbf{v}}; \boldsymbol{\mu})) &= \arg \min_{\bar{\mathbf{v}} \in \text{range}(\bar{\Phi}_r)} \|\bar{\mathbf{Z}}\bar{\mathbf{v}} - \bar{\mathbf{Z}}\bar{\mathbf{r}}(\hat{\mathbf{v}}; \boldsymbol{\mu})\|_2^2 = \bar{\Phi}_r(\bar{\mathbf{Z}}\bar{\Phi}_r)^+\bar{\mathbf{Z}}\bar{\mathbf{r}}(\hat{\mathbf{v}}; \boldsymbol{\mu}). \end{aligned}$$

As with space–time collocation, space–time gappy POD leads to hyper-reduction, as evaluating the objective function requires evaluating only $\bar{n}_z \leq N_s N_t$ elements of the spatiotemporal residual. It can also lead to a positive semidefinite metric $\Theta = \bar{\mathbf{A}}^T \bar{\mathbf{A}}$, as $\text{rank}(\bar{\mathbf{A}}^T \bar{\mathbf{A}}) = \bar{n}_r \leq N_s N_t$ in this case. Due to its close relationship to the original GNAT method—in which case $\mathbf{A} = (\mathbf{Z}\Phi_r)^+\mathbf{Z} \in \mathbb{R}^{n_r \times N_s}$ in problem (3.4)–(3.5), we refer to this approach as the space–time GNAT (ST-GNAT) method.

5. Computing method ingredients. This section describes particular methods for constructing the ingredients required for the ST-LSPG method, namely the space–time trial subspace $\mathcal{S}\mathcal{T}$; the sampling matrix $\bar{\mathbf{Z}}$ in the case of hyper-reduction; and the residual basis $\bar{\Phi}_r$ in the case of ST-GNAT.

5.1. Space–time trial subspace. We first assume that a set of training parameter instances $\mathcal{D}_{\text{train}} := \{\boldsymbol{\mu}_{\text{train}}^1, \dots, \boldsymbol{\mu}_{\text{train}}^{n_{\text{train}}}\} \subset \mathcal{D}$ has been defined (by, e.g., uniform sampling, Latin-hypercube sampling, greedy sampling) for which the FOM (2.1) has been solved numerically using a linear multistep method (2.2) to obtain state “snapshots” $\mathbf{x}(\cdot; \boldsymbol{\mu}_{\text{train}}^i) \in \mathbb{R}^{N_s}$, $i \in \mathbb{N}(n_{\text{train}})$. This data-collection process is referred to as the “offline” stage in model reduction.

Previous work on space–time model reduction constructed the space–time trial subspace simply as the span of these snapshots, i.e.,

$$\mathcal{S}\mathcal{T} = \mathbf{x}^0(\boldsymbol{\mu}) \otimes \mathcal{O} + \text{span}\{\mathbf{x}(\cdot; \boldsymbol{\mu}_{\text{train}}^i) - \mathbf{x}^0(\boldsymbol{\mu}_{\text{train}}^i)\}_{i=1}^{n_{\text{train}}} \subseteq \mathbb{R}^{N_s} \otimes \mathcal{H}$$

such that $\boldsymbol{\pi}_i = \mathbf{x}(\cdot; \boldsymbol{\mu}_{\text{train}}^i) - \mathbf{x}^0(\boldsymbol{\mu}_{\text{train}}^i) \in \mathbb{R}^{N_s} \otimes \mathcal{H}$, $i \in \mathbb{N}(n_{\text{train}})$ [45, 46, 50, 49, 17]. Unfortunately, because this approach extracts only a single space–time basis vector from each training simulation, it limits the dimension of the space–time ROM to $n_{st} = n_{\text{train}}$. In practical contexts where a single training simulation may incur significant computational costs, this can significantly limit the dimensionality (and resulting accuracy) of the space–time ROM. Further, storage costs can be significant in this case, as the basis requires $n_{st} N_s N_t$ storage.

To overcome these shortcomings, we propose to compute the space–time trial subspace by applying tensor-decomposition techniques to the three-way “state tensor” $\mathcal{X} \in \mathbb{R}^{N_s \times N_t \times n_{\text{train}}}$ with elements

$$\mathcal{X}_{ijk} := x_i(t^j; \boldsymbol{\mu}_{\text{train}}^k) - x_i^0(\boldsymbol{\mu}_{\text{train}}^k), \quad i \in \mathbb{N}(N_s), \quad j \in \mathbb{N}(N_t), \quad k \in \mathbb{N}(n_{\text{train}}).$$

The resulting space-time trial subspace comprises the direct sum of Kronecker products of spatial and temporal subspaces, i.e.,

$$\mathcal{S}\mathcal{T} = \mathbf{x}^0(\boldsymbol{\mu}) \otimes \mathcal{O} + \bigoplus_{i=1}^{n_s} \mathcal{S}_i \otimes \mathcal{T}_i,$$

where $\mathcal{S}_i := \text{span}\{\phi_i\} \subseteq \mathbb{R}^{N_s}$, $\mathcal{T}_i := \text{span}\{\psi_j^i\}_{j=1}^{n_t^i} \subseteq \mathcal{H}$, and $n_t^i \leq N_t$ (hopefully with $n_t^i \ll N_t$) for $i \in \mathbb{N}(n_s)$. Here, $\psi_j^i \in \mathcal{H}$ with $\psi_j^i(0) = 0$ denotes the j th temporal basis vector associated with the temporal behavior of i th spatial basis vector ϕ_i . Thus, in this case we have $\pi_{\mathcal{I}(i,j)} = \phi_i \psi_j^i \in \mathbb{R}^{N_s} \otimes \mathcal{H}$, $i \in \mathbb{N}(n_s)$, $j \in \mathbb{N}(n_t^i)$, and $n_{st} = \sum_{i=1}^{n_s} n_t^i$, where $\mathcal{I} : (i, j) \mapsto \sum_{k=1}^{i-1} n_t^k + j$ provides a mapping from the spatial-basis and temporal-basis indices to a space-time basis index. This approach enables a larger space-time ROM dimension, as $n_{st} = \sum_{i=1}^{n_s} n_t^i > n_{\text{train}}$ is possible; further, this approach requires only $n_s N_s + n_{st} N_t$ storage at most. We expect the resulting ST-LSPG ROM be accurate if the solution exhibits separable behavior in space and time.

The mode-1 and mode-2 unfolding of \mathcal{X} can be written as

$$\begin{aligned} \mathbf{X}_{(1)} &= [\mathbf{X}(\boldsymbol{\mu}_{\text{train}}^1) \quad \dots \quad \mathbf{X}(\boldsymbol{\mu}_{\text{train}}^{n_{\text{train}}})] \in \mathbb{R}^{N_s \times N_t n_{\text{train}}}, \\ \mathbf{X}_{(2)} &= [\mathbf{X}^T(\boldsymbol{\mu}_{\text{train}}^1) \quad \dots \quad \mathbf{X}^T(\boldsymbol{\mu}_{\text{train}}^{n_{\text{train}}})] \in \mathbb{R}^{N_t \times N_s n_{\text{train}}}, \end{aligned}$$

respectively, where we have defined $\mathbf{X}(\boldsymbol{\mu}) := \mathbf{g}(\mathbf{x}(\cdot; \boldsymbol{\mu}) - \mathbf{x}^0(\boldsymbol{\mu}))$. In the model-reduction literature, the matrix $\mathbf{X}_{(1)}$ is typically referred to as the “global snapshot matrix”; we refer to it in this work as the “spatial snapshot matrix,” as its columns comprise snapshots of the spatial solution over time and parameter variation. Similarly, we refer to $\mathbf{X}_{(2)}$ as the “temporal snapshot matrix,” as its columns comprise snapshots of the time-evolution of the solution over variation in space and parameter.

5.1.1. Spatial subspaces. We propose to compute the spatial subspaces $\mathcal{S}_i := \text{span}\{\phi_i\}$, $i \in \mathbb{N}(n_s)$, via proper orthogonal decomposition (POD) applied to the spatial snapshot matrix $\mathbf{X}_{(1)}$. In particular, we compute the spatial bases from the singular value decomposition (SVD) as

$$\begin{aligned} \mathbf{X}_{(1)} &= \mathbf{U}_s \boldsymbol{\Sigma}_s \mathbf{V}_s^T \in \mathbb{R}^{N_s \times N_t n_{\text{train}}}, \\ \phi_i &= \mathbf{u}_s^i, \quad i \in \mathbb{N}(n_s), \end{aligned}$$

where $n_s \leq \min(N_s, N_t n_{\text{train}})$ and $\mathbf{U}_s \equiv [\mathbf{u}_s^1 \quad \dots \quad \mathbf{u}_s^{N_t n_{\text{train}}}]$. The spatial subspace requires $n_s N_s$ storage. We now describe three approaches to computing the temporal subspaces \mathcal{T}_i , $i \in \mathbb{N}(n_s)$, from the state tensor \mathcal{X} .

5.1.2. Fixed temporal subspace via T-HOSVD. The most straightforward approach is to compute a fixed temporal subspace $\mathcal{T} := \text{span}\{\psi_i\}_{i=1}^{n_t} \subseteq \mathcal{H}$ such that $\mathcal{T}_i = \mathcal{T}$ and $n_t^i = n_t$, $i \in \mathbb{N}(n_s)$. We can compute such a subspace by applying POD to the temporal snapshot matrix $\mathbf{X}_{(2)}$. Specifically, we can compute the fixed temporal bases from the SVD of $\mathbf{X}_{(2)}$ as

$$(5.1) \quad \begin{aligned} \mathbf{X}_{(2)} &= \mathbf{U}_t \boldsymbol{\Sigma}_t \mathbf{V}_t^T \in \mathbb{R}^{N_t \times N_s n_{\text{train}}}, \\ \psi_j &= \mathbf{u}_t^j, \quad j \in \mathbb{N}(n_t), \end{aligned}$$

where $n_t \leq N_s n_{\text{train}}$, $\psi_j := \mathbf{h}(\psi_j)$, and $\mathbf{U}_t \equiv [\mathbf{u}_t^1 \quad \dots \quad \mathbf{u}_t^{N_s n_{\text{train}}}]$. This approach is equivalent to applying the truncated higher-order SVD (T-HOSVD) [18, 44] to the state tensor \mathcal{X} , and it requires $n_t N_t$ storage for the temporal subspace. We note that this approach is similar to that proposed in [6] in the context of space-time Galerkin projection performed at the time-continuous level.

5.1.3. Fixed temporal subspace via ST-HOSVD. Alternatively, we recall from (4.1) that each space-time basis vector is the Kronecker product of a spatial basis vector with a temporal basis vector. Thus, it is sensible to compute the temporal subspace according to the observed time evolution of the solution in the coordinates defined by the spatial subspace. Mathematically, we can achieve this by applying the sequentially truncated HOSVD (ST-HOSVD) [47, 2]. Rather than computing the SVD of $\mathbf{X}_{(2)}$, which is agnostic to dimensionality reduction in space, ST-HOSVD instead computes the SVD of the mode-2 unfolding of $\mathcal{X}(\Phi)$, where we have defined $\mathcal{X}(\mathbf{V}) := \mathcal{X} \times_1 \mathbf{V}$ for $\mathbf{V} \in \mathbb{R}^{N_s \times p}$ such that $\mathbf{X}(\mathbf{V})_{(1)} = \mathbf{V}^T \mathbf{X}_{(1)}$. In particular, we have

$$(5.2) \quad \mathbf{X}(\Phi)_{(2)} = \mathbf{U}_t(\Phi) \Sigma_t(\Phi) \mathbf{V}_t(\Phi)^T \in \mathbb{R}^{N_t \times n_s n_{\text{train}}},$$

$$(5.3) \quad \psi_j = \mathbf{u}_t^j(\Phi), \quad j \in \mathbb{N}(n_t),$$

where $n_t \leq n_s n_{\text{train}}$ and $\mathbf{U}_t(\Phi) \equiv [\mathbf{u}_t^1(\Phi) \cdots \mathbf{u}_t^{n_s n_{\text{train}}}(\Phi)]$. In addition to enabling the temporal basis to be associated with the time evolution of the spatial basis Φ , this approach is less computationally expensive than applying the T-HOSVD, as $\mathbf{X}(\Phi)_{(2)} \in \mathbb{R}^{N_t \times n_s n_{\text{train}}}$, while $\mathbf{X}_{(2)} \in \mathbb{R}^{N_t \times N_s n_{\text{train}}}$ and typically $n_s \ll N_s$. This temporal subspace also requires $n_t N_t$ storage.

5.1.4. Tailored temporal subspaces via ST-HOSVD. We can further tailor the temporal bases to capture the time evolution of each individual spatial basis vector. To achieve this using the ST-HOSVD, we compute the bases as

$$(5.4) \quad \mathbf{X}(\phi_i)_{(2)} = \mathbf{U}_t(\phi_i) \Sigma_t(\phi_i) \mathbf{V}_t(\phi_i)^T \in \mathbb{R}^{N_t \times n_{\text{train}}},$$

$$(5.5) \quad \psi_j^i = \mathbf{u}_t^j(\phi_i), \quad i \in \mathbb{N}(n_s), \quad j \in \mathbb{N}(n_t^i),$$

where $\psi_j^i := \mathbf{h}(\psi_j^i)$ and $\mathbf{U}_t(\phi_i) := [\mathbf{u}_t^1(\phi_i) \cdots \mathbf{u}_t^{n_{\text{train}}}(\phi_i)]$. This approach generates a tailored temporal subspace $\mathcal{T}_i := \text{span}\{\psi_j^i\}_{j=1}^{n_t^i}$ for each spatial subspace $\mathcal{S}_i := \text{span}\{\phi_i\}$. Further, because $\mathbf{X}(\phi_i)_{(2)} \in \mathbb{R}^{N_t \times n_{\text{train}}}$, $i \in \mathbb{N}(n_s)$, the cost of computing the n_s SVDs (5.4) is significantly less than computing the SVDs in either (5.1) or (5.2); this results from the quadratic dependence of the SVD cost on the number of columns in the matrix. However, the maximum dimension of each temporal basis is limited to the number of training-parameter instances, i.e., $n_t^i \leq n_{\text{train}}$, $i \in \mathbb{N}(n_s)$. This temporal subspace requires $\sum_{i=1}^{n_s} n_t^i N_t = n_{st} N_t$ storage, which is larger than that required by the fixed temporal subspaces. We note that this was the approach employed to construct temporal bases in our previous work based on forecasting [12, 10].

5.2. Space-time residual basis. We propose to construct the space-time residual basis $\bar{\Phi}_r \in \mathbb{V}_{\bar{n}_r}(\mathbb{R}^{N_s N_t})$ from training data comprising the space-time residual computed at a set of pairs of reduced coordinates and parameter instances, i.e., $\{\hat{\mathbf{y}}_{\text{res}}^i, \boldsymbol{\mu}_{\text{res}}^i\}_{i \in \mathbb{N}(n_{\text{res}})}$. In this case, the space-time residual “snapshots” can be expressed in a residual tensor $\mathcal{R} \in \mathbb{R}^{N_s \times N_t \times n_{\text{res}}}$ with entries

$$\mathcal{R}_{ijk} := r_i(t^j; \hat{\mathbf{y}}_{\text{res}}^k; \boldsymbol{\mu}_{\text{res}}^k).$$

We propose three methods for determining these training instances $\{\hat{\mathbf{y}}_{\text{res}}^i, \boldsymbol{\mu}_{\text{res}}^i\}_{i \in \mathbb{N}(n_{\text{res}})}$.

1. *ST-LSPG ROM training iterations.* This approach employs

$$\{\hat{\mathbf{y}}_{\text{res}}^i, \boldsymbol{\mu}_{\text{res}}^i\}_{i \in \mathbb{N}(n_{\text{res}})} = \{\hat{\mathbf{y}}^{(k)}(\boldsymbol{\mu}), \boldsymbol{\mu}\}_{\boldsymbol{\mu} \in \mathcal{D}_{\text{res}}, k \in \{0, \dots, k_{\max}(\boldsymbol{\mu})\}},$$

where $\hat{\mathbf{y}}^{(k)}(\boldsymbol{\mu})$ corresponds to the ST-LSPG solution at the k th Gauss-Newton iteration (4.7)–(4.8) for some specified weighting matrix $\bar{\mathbf{A}}$ that does not rely on data (e.g., $\bar{\mathbf{A}} = \mathbf{I}_{N_s N_t}$), and $\mathcal{D}_{\text{res}} \subset \mathcal{D}$ denotes a set of training parameter instances that is in general different from $\mathcal{D}_{\text{train}}$. This case leads to $n_{\text{res}} = \sum_{\boldsymbol{\mu} \in \mathcal{D}_{\text{res}}} (k_{\max}(\boldsymbol{\mu}) + 1)$ and requires $|\mathcal{D}_{\text{res}}|$ training simulations of the ST-LSPG ROM.

2. *Projection of FOM training solutions.* This approach employs

$$\{\hat{\mathbf{y}}_{\text{res}}^i, \boldsymbol{\mu}_{\text{res}}^i\}_{i \in \mathbb{N}(n_{\text{res}})} = \{\hat{\mathbf{x}}(\boldsymbol{\mu}), \boldsymbol{\mu}\}_{\boldsymbol{\mu} \in \mathcal{D}_{\text{res}}},$$

where $\hat{\mathbf{x}}(\boldsymbol{\mu})$ is defined as

$$(5.6) \quad \hat{\mathbf{x}}(\boldsymbol{\mu}) := (\boldsymbol{\Pi}^T \boldsymbol{\Pi})^{-1} \boldsymbol{\Pi}^T \mathbf{h}(\mathbf{x}(\cdot; \boldsymbol{\mu}) - \mathbf{x}^0(\boldsymbol{\mu})),$$

where $\boldsymbol{\Pi} := [\pi_1 \ \cdots \ \pi_{n_{st}}] \in \mathbb{R}^{N_s N_t \times n_{st}}$. This approach does not require any additional training simulations; it simply requires n_{res} evaluations of the space-time residual.

3. *Random samples.* In this approach, the training set $\{\hat{\mathbf{y}}_{\text{res}}^i, \boldsymbol{\mu}_{\text{res}}^i\}_{i \in \mathbb{N}(n_{\text{res}})}$ comprises random samples (e.g., via Latin hypercube sampling) from $\mathcal{D}_{\hat{\mathbf{y}}} \times \mathcal{D}$ where $\mathcal{D}_{\hat{\mathbf{y}}} \subseteq \mathbb{R}^{n_{st}}$. This approach also requires only n_{res} evaluations of the space-time residual.

Given the residual tensor \mathcal{R} , we can compute the associated space-time residual basis in a manner analogous to the approaches proposed in section 5.1. That is, we can compute spatial residual bases as

$$\begin{aligned} \mathbf{R}_{(1)} &= \mathbf{U}_{r,s} \boldsymbol{\Sigma}_{r,s} \mathbf{V}_{r,s}^T \in \mathbb{R}^{N_s \times N_t n_{\text{res}}}, \\ \phi_{r,i} &= \mathbf{u}_{r,s}^i, \quad i \in \mathbb{N}(n_{r,s}), \end{aligned}$$

with $n_{r,s} \leq N_t n_{\text{res}}$ and temporal residual bases via either the T-HOSVD

$$\begin{aligned} \mathbf{R}_{(2)} &= \mathbf{U}_{r,t} \boldsymbol{\Sigma}_{r,t} \mathbf{V}_{r,t}^T \in \mathbb{R}^{N_t \times N_s n_{\text{res}}}, \\ \psi_{r,j} &= \mathbf{u}_{r,t}^j, \quad j \in \mathbb{N}(n_{r,t}), \end{aligned}$$

with $n_{r,t} \leq N_s n_{\text{res}}$, the ST-HOSVD

$$\begin{aligned} \mathbf{R}(\Phi_r)_{(2)} &= \mathbf{U}_{r,t}(\Phi_r) \boldsymbol{\Sigma}_{r,t}(\Phi_r) \mathbf{V}_{r,t}(\Phi_r)^T \in \mathbb{R}^{N_t \times n_{r,s} n_{\text{res}}}, \\ \psi_{r,j} &= \mathbf{u}_{r,t}^j(\Phi_r), \quad j \in \mathbb{N}(n_{r,t}), \end{aligned}$$

with $n_{r,t} \leq n_{r,s} n_{\text{res}}$, or the tailored ST-HOSVD

$$(5.7) \quad \mathbf{R}(\phi_{r,i})_{(2)} = \mathbf{U}_{r,t}(\phi_{r,i}) \boldsymbol{\Sigma}_{r,t}(\phi_{r,i}) \mathbf{V}_{r,t}(\phi_{r,i})^T \in \mathbb{R}^{N_t \times n_{\text{res}}}, \quad i \in \mathbb{N}(n_{r,s}),$$

$$(5.8) \quad \psi_{r,j}^i = \mathbf{u}_{r,t}^i(\phi_{r,i}), \quad i \in \mathbb{N}(n_{r,s}), \quad j \in \mathbb{N}(n_{r,t}^i),$$

with $n_{r,t}^i \leq n_{\text{res}}$, where $\mathcal{R}(\mathbf{V}) := \mathcal{R} \times_1 \mathbf{V}$, $\psi_{r,j} := \mathbf{h}(\psi_{r,j})$, and $\psi_{r,j}^i := \mathbf{h}(\psi_{r,j}^i)$.

We compute the orthogonal residual basis $\bar{\Phi}_r \in \mathbb{V}_{\bar{n}_r}(\mathbb{R}^{N_s N_t})$ from the QR factorization of the nonorthogonalized basis $\boldsymbol{\Pi}_r \equiv [\pi_{r,1} \ \cdots \ \pi_{r,\bar{n}_r}] \in \mathbb{R}^{N_s N_t \times \bar{n}_r}$ as

$$\boldsymbol{\Pi}_r = \bar{\Phi}_r \mathbf{R},$$

where $\pi_{r,\mathcal{J}_r(i,j)} = \mathbf{h}(\phi_{r,i} \otimes \psi_{r,j}^i)$ and $\mathcal{J}_r : (i, j) \mapsto \sum_{k=1}^{i-1} n_{r,t}^k + j$ provides a mapping from the spatial-basis and temporal-basis indices to a space-time basis index for the residual.

5.3. Sampling matrix. We propose three approaches to computing the space–time sample set $\mathbf{st} := \{(\mathbf{s}_i, t_i)\}_{i \in \mathbb{N}(\bar{n}_z)}$ that defines the residual-sampling matrix $\bar{\mathbf{Z}}$ in (4.10).

1. *Greedy sampling of space–time indices.* This approach selects space–time indices in a greedy manner by executing Algorithm 1, which is a space–time adaptation of the greedy method presented in [9, 11] that allows for oversampling to enable least-squares regression via gappy POD.
2. *Sequential greedy sampling of spatial then temporal indices.* This approach computes space–time sample indices as the Cartesian product of spatial and temporal samples, i.e., $\mathbf{st} = \mathbf{s} \times \mathbf{t}$. First, the approach selects spatial indices \mathbf{s} by executing Algorithm 3 with inputs $\bar{\Phi}_r$, the desired number of spatial samples \bar{n}_s , and $\mathbf{t} = \mathbb{N}(N_t)$ (i.e., full temporal sampling). Then, the method selects temporal indices \mathbf{t} by executing Algorithm 2 with inputs $\bar{\Phi}_r$, the desired number of temporal samples \bar{n}_t , and \mathbf{s} computed from Algorithm 3.
3. *Sequential greedy sampling of temporal then spatial indices.* This approach also computes space–time sample indices as $\mathbf{st} = \mathbf{s} \times \mathbf{t}$. First, the approach selects temporal indices \mathbf{t} by executing Algorithm 2 with inputs $\bar{\Phi}_r$, the desired number of temporal samples \bar{n}_t , and $\mathbf{s} = \mathbb{N}(N_s)$ (i.e., full spatial sampling). Then, the method selects spatial indices \mathbf{s} by executing Algorithm 3 with inputs $\bar{\Phi}_r$, the desired number of spatial samples \bar{n}_s , and \mathbf{t} computed from Algorithm 2.

We note that enforcing $\mathbf{st} = \mathbf{s} \times \mathbf{t}$ as in approaches 2 and 3 above comes with a practical advantage. Namely, a single sample mesh [11]—which is tasked with computing spatial samples associated with \mathbf{s} —can be employed for all sampled time instances \mathbf{t} .

5.4. Initial guess. One practical challenge of ST-LSPG relative to (spatial-projection-based) LSPG is devising an accurate initial guess $\hat{\mathbf{y}}^{(0)}$ for the Gauss–Newton iterations (4.7)–(4.8). In LSPG, the initial guess employed when solving problem (3.4)–(3.5) at a given time instance t^n using the Gauss–Newton method can be set to the solution from the previous time instance, i.e., $\hat{\mathbf{x}}(t^{n(0)}; \boldsymbol{\mu}) = \hat{\mathbf{x}}(t^{n-1}; \boldsymbol{\mu})$. This choice typically leads to rapid convergence due to the fact that the state undergoes limited variation between time instances, particularly for small time steps Δt^n . Alternatively, accurate initial guesses based on polynomial extrapolation or forecasting [12] can be employed to further improve convergence.

However, in ST-LSPG, deriving an accurate initial guess $\hat{\mathbf{y}}^{(0)}(\boldsymbol{\mu})$ is less straightforward. We propose computing $\hat{\mathbf{y}}^{(0)}(\boldsymbol{\mu})$ as an interpolant of the generalized coordinates $\hat{\mathbf{y}}(\boldsymbol{\mu})$ in the parameter space. That is, given the training parameter instances $\mathcal{D}_{\text{train}} \subset \mathcal{D}$ for which the FOM has been solved, we can compute the projection of these solutions onto the space–time trial subspace as $\{\hat{\mathbf{x}}(\boldsymbol{\mu})\}_{\boldsymbol{\mu} \in \mathcal{D}_{\text{train}}}$ with $\hat{\mathbf{x}}(\boldsymbol{\mu})$ defined in (5.6). Then, we can compute $\hat{\mathbf{y}}^{(0)}(\boldsymbol{\mu})$ via interpolation (or least-squares regression) in the parameter space \mathcal{D} using data $\{\hat{\mathbf{x}}(\boldsymbol{\mu})\}_{\boldsymbol{\mu} \in \mathcal{D}_{\text{train}}}$.

6. Error analysis. For simplicity, this section omits parameter dependence of all operators and assumes a uniform time step, i.e., $\Delta t^n = \Delta t$, $n \in \mathbb{N}(N_t)$. Thus, the FOM solution \mathbf{x} and the ST-LSPG ROM solution $\tilde{\mathbf{y}}$ satisfy

$$(6.1) \quad \mathbf{r}(\cdot; \mathbf{x}) = \mathbf{0} \quad \text{and} \quad \tilde{\mathbf{y}} = \arg \min_{\mathbf{w} \in \mathcal{S}^{\mathcal{T}}} \|\mathbf{r}(\cdot; \mathbf{w})\|_{\bar{\mathbf{A}}^T \bar{\mathbf{A}}},$$

respectively. We begin by stating assumptions that will be leveraged in subsequent analyses.

Algorithm 1. Greedy algorithm for constructing spatiotemporal sample set \mathbf{st} .

Input: residual basis $\bar{\Phi}_r \in \mathbb{V}_{\bar{n}_r}(\mathbb{R}^{N_s N_t})$; desired number of spatial samples $\bar{n}_z \leq N_s N_t$

Output: space-time sample set $\mathbf{st} \subseteq \mathbb{N}(N_s) \times \mathbb{N}(N_t)$

- 1: $\mathbf{st} \leftarrow \emptyset$ {Initialize spatiotemporal sample set.}
- 2: Determine number of spatiotemporal samples to compute at each greedy iteration:

$$\bar{n}_z^i = \begin{cases} \text{floor}(\bar{n}_z / \bar{n}_r) + 1, & i = 1, \dots, \bar{n}_z \bmod \bar{n}_r, \\ \text{floor}(\bar{n}_z / \bar{n}_r), & i = \bar{n}_z \bmod \bar{n}_r + 1, \dots, \bar{n}_r, \end{cases}$$

- 3: **for** $i = 1, \dots, \bar{n}_r$ **do** {Greedy iteration}
 - 4: **if** $i = 1$ **then**
 - 5: $\boldsymbol{\varepsilon} \leftarrow \bar{\phi}_{r,1}$ {Initialize the error vector.}
 - 6: **else**
 - 7: $\boldsymbol{\varepsilon} \leftarrow \left(\mathbf{I}_{N_s N_t} - [\bar{\phi}_{r,1} \ \dots \ \bar{\phi}_{r,i-1}] (\bar{\mathbf{Z}} [\bar{\phi}_{r,1} \ \dots \ \bar{\phi}_{r,i-1}])^+ \bar{\mathbf{Z}} \right) \bar{\phi}_{r,i}$, where $\bar{\mathbf{Z}}$ is defined in (4.10). {Compute the error in the gappy POD approximation of $\bar{\phi}_{r,i}$ }
 - 8: **end if**
 - 9: **for** $j = 1, \dots, \bar{n}_z^i$ **do**
 - 10: $(\mathbf{z}^*, t^*) = \arg \max_{(k,n) \in \mathbb{N}(N_s) \times \mathbb{N}(N_t) \setminus \mathbf{st}} |\varepsilon_k(t^n)|$, where $\boldsymbol{\varepsilon} \equiv [\varepsilon_1 \ \dots \ \varepsilon_{N_s}]^T := \mathbf{h}^{-1}(\boldsymbol{\varepsilon})$
 {Identify the spatiotemporal index with the largest gappy POD error.}
 - 11: $\mathbf{st} \leftarrow \mathbf{st} \cup \{(\mathbf{z}^*, t^*)\}$ {Include the identified space-time index in the spatiotemporal sample set.}
 - 12: **end for**
 - 13: **end for**
-

A1 There exists a constant $L_f > 0$ such that

$$\|\mathbf{f}(\mathbf{w}, t) - \mathbf{f}(\mathbf{y}, t)\|_2 \leq L_f \|\mathbf{w} - \mathbf{y}\|_2 \quad \forall \mathbf{w}, \mathbf{y} \in \mathbb{R}^{N_s}, \quad \forall t \in [0, T].$$

A2 The time step Δt is sufficiently small such that

$$\Delta t < \frac{\sigma_{\min}(\mathbf{A}_{LM})}{L_f \sigma_{\max}(\mathbf{B}_{LM})},$$

where

$$\mathbf{A}_{LM} := \begin{bmatrix} \alpha_0^1 \mathbf{I} & & & \\ \alpha_1^2 \mathbf{I} & \alpha_0^2 \mathbf{I} & & \\ & \ddots & \ddots & \\ & & \alpha_{k(t^{N_t})}^{N_t} \mathbf{I} & \cdots & \alpha_0^{N_t} \mathbf{I} \end{bmatrix}, \quad \mathbf{B}_{LM} := \begin{bmatrix} \beta_0^1 \mathbf{I} & & & \\ \beta_1^2 \mathbf{I} & \beta_0^2 \mathbf{I} & & \\ & \ddots & \ddots & \\ & & \beta_{k(t^{N_t})}^{N_t} \mathbf{I} & \cdots & \beta_0^{N_t} \mathbf{I} \end{bmatrix},$$

where $\mathbf{I} = \mathbf{I}_{N_s}$ here and $\sigma_{\min}(\mathbf{A})$ and $\sigma_{\max}(\mathbf{A})$ denote the minimum and maximum singular values of the matrix \mathbf{A} , respectively.

A3 The space-time weighting matrix $\bar{\mathbf{A}}$ is defined such that the residual in the weighted space-time norm is uniformly bounded below by the ℓ^2 -norm of the residual over all elements of the space-time trial subspace; i.e., there exists $P > 0$ such that

$$\|\bar{\mathbf{A}}\mathbf{r}(\cdot; \mathbf{w})\|_2 \geq P \|\mathbf{r}(\cdot; \mathbf{w})\|_2 \quad \forall \mathbf{w} \in \mathcal{ST}.$$

Algorithm 2. Greedy algorithm for constructing temporal sample set \mathbf{t} .

Input: residual basis $\bar{\Phi}_r \in \mathbb{V}_{\bar{n}_r}(\mathbb{R}^{N_s N_t})$; desired number of temporal samples $\bar{n}_t \leq N_t$; spatial sample set $\mathbf{s} \subseteq \mathbb{N}(N_s)$

Output: temporal sample set $\mathbf{t} \subseteq \mathbb{N}(N_t)$

- 1: $\mathbf{t} \leftarrow \emptyset$ {Initialize temporal sample set.}
- 2: Determine number of temporal samples to compute at each greedy iteration:

$$\bar{n}_t^i = \begin{cases} \text{floor}(\bar{n}_t / \bar{n}_r) + 1, & i = 1, \dots, \bar{n}_t \bmod \bar{n}_r, \\ \text{floor}(\bar{n}_t / \bar{n}_r), & i = \bar{n}_t \bmod \bar{n}_r + 1, \dots, \bar{n}_r, \end{cases}$$

- 3: **for** $i = 1, \dots, \bar{n}_r$ **do** {Greedy iteration}
 - 4: **if** $i = 1$ **then**
 - 5: $\boldsymbol{\varepsilon} \leftarrow \bar{\phi}_{r,1}$ {Initialize the error vector.}
 - 6: **else**
 - 7: $\boldsymbol{\varepsilon} \leftarrow \left(\mathbf{I}_{N_s N_t} - [\bar{\phi}_{r,1} \ \cdots \ \bar{\phi}_{r,i-1}] (\bar{\mathbf{Z}} [\bar{\phi}_{r,1} \ \cdots \ \bar{\phi}_{r,i-1}])^+ \bar{\mathbf{Z}} \right) \bar{\phi}_{r,i}$, where
 $\bar{\mathbf{Z}}$ is defined in (4.10) with $\mathbf{s}\mathbf{t} = \mathbf{s} \times \mathbf{t}$.
{Compute the error in the gappy POD approximation of $\bar{\phi}_{r,i}$.}
 - 8: **end if**
 - 9: **for** $j = 1, \dots, \bar{n}_t^i$ **do**
 - 10: $t^* = \arg \max_{n \in \mathbb{N}(N_t) \setminus \mathbf{t}} \|\boldsymbol{\varepsilon}(t^n)\|_2^2$, where $\boldsymbol{\varepsilon} := \mathbf{h}^{-1}(\boldsymbol{\varepsilon})$
{Identify the temporal index with the largest gappy POD error averaged over all spatial indices.}
 - 11: $\mathbf{t} \leftarrow \mathbf{t} \cup \{t^*\}$ {Include the identified temporal index in the temporal sample set.}
 - 12: **end for**
 - 13: **end for**
-

LEMMA 6.1. Under assumption A1, the linear multistep residual is also Lipschitz continuous, i.e.,

$$\|\mathbf{r}(\cdot; \mathbf{w}) - \mathbf{r}(\cdot; \mathbf{y})\|_{\bar{\mathbf{A}}^T \bar{\mathbf{A}}} \leq L_r \|\mathbf{w} - \mathbf{y}\|_2 \quad \forall \mathbf{w}, \mathbf{y} \in \mathbb{R}^{N_s} \otimes \mathcal{H},$$

with Lipschitz constant

$$(6.2) \quad L_r := \sigma_{\max}(\bar{\mathbf{A}} \mathbf{A}_{LM}) + \Delta t L_f \sigma_{\max}(\bar{\mathbf{A}} \mathbf{B}_{LM}).$$

Proof. Defining $\bar{\mathbf{f}} : \mathbf{w} \mapsto \mathbf{h}(\mathbf{f}(\mathbf{w}(\cdot), \cdot))$ and noting that $\|\bar{\mathbf{f}}(\mathbf{w})\|_2^2 = \sum_{n=1}^{N_t} \|\mathbf{f}(\mathbf{w}(t^n), t^n)\|_2^2 \leq L_f^2 \sum_{n=1}^{N_t} \|\mathbf{w}(t^n)\|_2^2 = L_f^2 \|\mathbf{w}\|_2^2$, we have

$$\|\bar{\mathbf{f}}(\mathbf{w})\|_2 \leq L_f \|\mathbf{w}\|_2 \quad \forall \mathbf{w} \in \mathbb{R}^{N_s} \otimes \mathcal{H};$$

i.e., the Lipschitz constant of $\bar{\mathbf{f}}$ is identical to that of \mathbf{f} . Further noting that

$$\bar{\mathbf{r}}(\mathbf{w}) = \mathbf{A}_{LM} \mathbf{h}(\mathbf{w}) - \Delta t \mathbf{B}_{LM} \bar{\mathbf{f}}(\mathbf{w}) + \mathbf{h}(\mathbf{b}),$$

where $\mathbf{b}(t^n) = \alpha_n^n \mathbf{x}^0 - \Delta t \beta_n^n \mathbf{f}(\mathbf{x}^0)$, we have from the triangle inequality

$$\begin{aligned} \|\mathbf{r}(\cdot; \mathbf{w}) - \mathbf{r}(\cdot; \mathbf{y})\|_{\bar{\mathbf{A}}^T \bar{\mathbf{A}}} &= \|\bar{\mathbf{A}}(\bar{\mathbf{r}}(\mathbf{w}) - \bar{\mathbf{r}}(\mathbf{y}))\|_2 \\ &= \|\bar{\mathbf{A}} \mathbf{A}_{LM}(\mathbf{h}(\mathbf{w}) - \mathbf{h}(\mathbf{y})) - \Delta t \bar{\mathbf{A}} \mathbf{B}_{LM}(\bar{\mathbf{f}}(\mathbf{w}) - \bar{\mathbf{f}}(\mathbf{y}))\|_2 \\ &\leq (\sigma_{\max}(\bar{\mathbf{A}} \mathbf{A}_{LM}) + \Delta t L_f \sigma_{\max}(\bar{\mathbf{A}} \mathbf{B}_{LM})) \|\mathbf{w} - \mathbf{y}\|_2 \quad \forall \mathbf{w}, \mathbf{y} \in \mathbb{R}^{N_s} \otimes \mathcal{H}. \quad \square \end{aligned}$$

Algorithm 3. Greedy algorithm for constructing spatial sample set \mathfrak{s} .

Input: residual basis $\bar{\Phi}_r \in \mathbb{V}_{\bar{n}_r}(\mathbb{R}^{N_s N_t})$; desired number of spatial samples $\bar{n}_s \leq N_s$; temporal sample set $\mathbf{t} \subseteq \mathbb{N}(N_t)$

Output: spatial sample set $\mathfrak{s} \subseteq \mathbb{N}(N_s)$

- 1: $\mathfrak{s} \leftarrow \emptyset$ {Initialize spatial sample set.}
- 2: Determine number of spatial samples to compute at each greedy iteration:

$$\bar{n}_s^i = \begin{cases} \text{floor}(\bar{n}_s / \bar{n}_r) + 1, & i = 1, \dots, \bar{n}_s \bmod \bar{n}_r, \\ \text{floor}(\bar{n}_s / \bar{n}_r), & i = \bar{n}_s \bmod \bar{n}_r + 1, \dots, \bar{n}_r, \end{cases}$$

- 3: **for** $i = 1, \dots, \bar{n}_r$ **do** {Greedy iteration}
- 4: **if** $i = 1$ **then**
- 5: $\boldsymbol{\varepsilon} \leftarrow \bar{\phi}_{r,1}$ {Initialize the error vector.}
- 6: **else**
- 7: $\boldsymbol{\varepsilon} \leftarrow (\mathbf{I}_{N_s N_t} - [\bar{\phi}_{r,1} \ \dots \ \bar{\phi}_{r,i-1}] (\bar{\mathbf{Z}} [\bar{\phi}_{r,1} \ \dots \ \bar{\phi}_{r,i-1}])^+ \bar{\mathbf{Z}}) \bar{\phi}_{r,i}$, where $\bar{\mathbf{Z}}$ is defined in (4.10) with $\mathbf{s}\mathbf{t} = \mathfrak{s} \times \mathbf{t}$. {Compute the error in the gappy POD approximation of $\bar{\phi}_{r,i}$ }
- 8: **end if**
- 9: **for** $j = 1, \dots, \bar{n}_s^i$ **do**
- 10: $\mathfrak{s}^* = \arg \max_{k \in \mathbb{N}(N_s) \setminus \mathfrak{s}} \sum_{n=1}^{N_t} (\varepsilon_k(t^n))^2$, where $\boldsymbol{\varepsilon} \equiv [\varepsilon_1 \ \dots \ \varepsilon_{N_s}]^T := \mathbf{h}^{-1}(\boldsymbol{\varepsilon})$ {Identify the spatial index with the largest gappy POD error averaged over all temporal indices.}
- 11: $\mathfrak{s} \leftarrow \mathfrak{s} \cup \{\mathfrak{s}^*\}$ {Include the identified spatial index in the spatial sample set.}
- 12: **end for**
- 13: **end for**

LEMMA 6.2. Under assumptions A1 and A2, the linear multistep residual is also inverse Lipschitz continuous, i.e.,

$$\|\mathbf{r}(\cdot; \mathbf{w}) - \mathbf{r}(\cdot; \mathbf{y})\|_2 \geq K_r \|\mathbf{w} - \mathbf{y}\|_2 \quad \forall \mathbf{w}, \mathbf{y} \in \mathbb{R}^{N_s} \otimes \mathcal{H},$$

with inverse Lipschitz constant

$$(6.3) \quad K_r := \sigma_{\min}(\mathbf{A}_{LM}) - \Delta t L_f \sigma_{\max}(\mathbf{B}_{LM}).$$

Proof. Applying the reverse triangle inequality and employing assumption A2 yields

$$\begin{aligned} \|\mathbf{r}(\cdot; \mathbf{w}) - \mathbf{r}(\cdot; \mathbf{y})\|_2 &= \|\mathbf{A}_{LM}(\mathbf{h}(\mathbf{w}) - \mathbf{h}(\mathbf{y})) - \Delta t \mathbf{B}_{LM}(\bar{\mathbf{f}}(\mathbf{w}) - \bar{\mathbf{f}}(\mathbf{y}))\|_2 \\ &\geq \|\mathbf{A}_{LM}(\mathbf{h}(\mathbf{w}) - \mathbf{h}(\mathbf{y}))\|_2 - \Delta t \|\mathbf{B}_{LM}(\bar{\mathbf{f}}(\mathbf{w}) - \bar{\mathbf{f}}(\mathbf{y}))\|_2, \end{aligned}$$

which directly leads to the desired result. \square

THEOREM 6.3 (a priori error bound with respect to ℓ^2 -optimal solution). Under assumptions A1, A2, and A3, the error in the ST-LSPG solution at any time instance can be bounded by the best approximation error as

$$\|\mathbf{x} - \tilde{\mathbf{y}}\|_2 \leq \frac{1}{P} \left(\frac{\sigma_{\max}(\bar{\mathbf{A}} \mathbf{A}_{LM}) + \Delta t L_f \sigma_{\max}(\bar{\mathbf{A}} \mathbf{B}_{LM})}{\sigma_{\min}(\mathbf{A}_{LM}) - \Delta t L_f \sigma_{\max}(\mathbf{B}_{LM})} \right) \min_{\mathbf{w} \in \mathcal{S}^T} \|\mathbf{x} - \mathbf{w}\|_2.$$

Proof. We begin by defining the ℓ^2 -optimal solution \mathbf{x}_2 , which satisfies

$$(6.4) \quad \mathbf{x}_2 = \arg \min_{\mathbf{w} \in \mathcal{ST}} \|\mathbf{x} - \mathbf{w}\|_2.$$

Then, we can exploit the optimality properties of \mathbf{x}_2 and $\tilde{\mathbf{y}}$ from (6.4) and (6.1), respectively; Lipschitz continuity (Lemma 6.1); residual-norm equivalence (assumption A3); and inverse Lipschitz continuity of the residual (Lemma 6.2) to derive the following inequalities:

$$\begin{aligned} \min_{\mathbf{w} \in \mathcal{ST}} \|\mathbf{x} - \mathbf{w}\|_2 &= \|\mathbf{x} - \mathbf{x}_2\|_2 \geq \frac{1}{L_r} \|\mathbf{r}(\cdot; \mathbf{x}_2)\|_{\bar{\mathbf{A}}^T \bar{\mathbf{A}}} \geq \min_{\mathbf{w} \in \mathcal{ST}} \frac{1}{L_r} \|\mathbf{r}(\cdot; \mathbf{w})\|_{\bar{\mathbf{A}}^T \bar{\mathbf{A}}} \\ &= \frac{1}{L_r} \|\mathbf{r}(\cdot; \tilde{\mathbf{y}})\|_{\bar{\mathbf{A}}^T \bar{\mathbf{A}}} \geq \frac{P}{L_r} \|\mathbf{r}(\cdot; \tilde{\mathbf{y}})\|_2 \geq \frac{PK_r}{L_r} \|\mathbf{x} - \tilde{\mathbf{y}}\|_2. \end{aligned}$$

Substituting in the definitions of L_r and K_r from (6.2) and (6.3), respectively, yields the stated result. \square

THEOREM 6.4 (a priori error bound with respect to ℓ^∞ -optimal solution). *Under assumptions A1, A2, and A3, the error in the ST-LSPG solution at any time instance can be bounded by the best approximation error as*

$$\begin{aligned} &\max_{n \in \mathbb{N}(N_t)} \|\mathbf{x}(t^n) - \tilde{\mathbf{y}}(t^n)\|_2 \\ &\leq \frac{\sqrt{N_t}}{P} \left(\frac{\sigma_{\max}(\bar{\mathbf{A}}\mathbf{A}_{LM}) + \Delta t L_f \sigma_{\max}(\bar{\mathbf{A}}\mathbf{B}_{LM})}{\sigma_{\min}(\mathbf{A}_{LM}) - \Delta t L_f \sigma_{\max}(\mathbf{B}_{LM})} \right) \min_{\mathbf{w} \in \mathcal{ST}} \max_{n \in \mathbb{N}(N_t)} \|\mathbf{x}(t^n) - \mathbf{w}(t^n)\|_2. \end{aligned}$$

Proof. We begin by defining the ℓ^∞ -optimal solution \mathbf{x}_∞ , which satisfies

$$(6.5) \quad \mathbf{x}_\infty = \arg \min_{\mathbf{w} \in \mathcal{ST}} \|\mathbf{x} - \mathbf{w}\|_\infty,$$

where we have defined the ℓ^∞ -norm as $\|\mathbf{w}\|_\infty := \max_{n \in \mathbb{N}(N_t)} \|\mathbf{w}(t^n)\|_2$. Then, we can exploit norm equivalence $\|\mathbf{w}\|_\infty \leq \|\mathbf{w}\|_2 \leq \sqrt{n} \|\mathbf{w}\|_\infty$ for $\mathbf{w} \in \mathbb{R}^n$; Lipschitz continuity (Lemma 6.1); residual-norm equivalence (assumption A3); and inverse Lipschitz continuity of the residual (Lemma 6.2); and the optimality properties of \mathbf{x}_∞ and $\tilde{\mathbf{y}}$ from (6.5) and (6.1), respectively, to derive the following inequalities:

$$\begin{aligned} \min_{\mathbf{w} \in \mathcal{ST}} \|\mathbf{x} - \mathbf{w}\|_\infty &= \|\mathbf{x} - \mathbf{x}_\infty\|_\infty \geq \frac{1}{\sqrt{N_t}} \|\mathbf{x} - \mathbf{x}_\infty\|_2 \geq \frac{1}{L_r \sqrt{N_t}} \|\mathbf{r}(\cdot; \mathbf{x}_\infty)\|_{\bar{\mathbf{A}}^T \bar{\mathbf{A}}} \\ &\geq \min_{\mathbf{w} \in \mathcal{ST}} \frac{1}{L_r \sqrt{N_t}} \|\mathbf{r}(\cdot; \mathbf{w})\|_{\bar{\mathbf{A}}^T \bar{\mathbf{A}}} \\ &= \frac{1}{L_r \sqrt{N_t}} \|\mathbf{r}(\cdot; \tilde{\mathbf{y}})\|_{\bar{\mathbf{A}}^T \bar{\mathbf{A}}} \geq \frac{P}{L_r \sqrt{N_t}} \|\mathbf{r}(\cdot; \tilde{\mathbf{y}})\|_2 \geq \frac{PK_r}{L_r \sqrt{N_t}} \|\mathbf{x} - \tilde{\mathbf{y}}\|_2 \\ &\geq \frac{PK_r}{L_r \sqrt{N_t}} \|\mathbf{x} - \tilde{\mathbf{y}}\|_\infty. \end{aligned}$$

Noting that $\|\mathbf{x} - \tilde{\mathbf{y}}\|_\infty \geq \|\mathbf{x}(t^n) - \tilde{\mathbf{y}}(t^n)\|_2 \forall n \in \mathbb{N}(N_t)$ and substituting in the definitions of L_r and K_r from (6.2) and (6.3), respectively, yields the stated result. \square

We now provide simplified variants of these error bounds in the case of unweighted LSPG (section 4.3.1) for which $\bar{\mathbf{A}}^n = \mathbf{I}_{N_s}$.

COROLLARY 6.5 (simplified a priori error bound). *If $\bar{\mathbf{A}} = \mathbf{I}_{N_s N_t}$, then under assumptions A1 and A2, the error in the ST-LSPG solution at any time instance can be bounded by the best approximation error as*

$$(6.6) \quad \|\mathbf{x} - \tilde{\mathbf{y}}\|_2 \leq (1 + \Lambda) \min_{\mathbf{w} \in \mathcal{ST}} \|\mathbf{x} - \mathbf{w}\|_2,$$

$$(6.7) \quad \max_{n \in \mathbb{N}(N_t)} \|\mathbf{x}(t^n) - \tilde{\mathbf{y}}(t^n)\|_2 \leq \sqrt{N_t} (1 + \Lambda) \min_{\mathbf{w} \in \mathcal{ST}} \max_{n \in \mathbb{N}(N_t)} \|\mathbf{x}(t^n) - \mathbf{w}(t^n)\|_2,$$

where we define the Lebesgue constant for a given time integrator and time step Δt as

$$\Lambda := \frac{\sigma_{\max}(\mathbf{A}_{LM}) - \sigma_{\min}(\mathbf{A}_{LM}) + 2\Delta t L_f \sigma_{\max}(\mathbf{B}_{LM})}{\sigma_{\min}(\mathbf{A}_{LM}) - \Delta t L_f \sigma_{\max}(\mathbf{B}_{LM})}.$$

Proof. The proof follows trivially from Theorems 6.3 and 6.4 by substituting $\bar{\mathbf{A}} = \mathbf{I}_{N_s N_t}$ and noting that assumption A3 is automatically satisfied for this choice of weighting matrix $\bar{\mathbf{A}}$, as $P = 1$ in this case. \square

Remark 6.1 (stability-constant growth). Figure 1 plots the dependence of the stability constants in the a priori error bounds (6.6) and (6.7) as a function of the final time T for multiple linear multistep methods and fixed values of the time step and Lipschitz constant. Critically, note that the stability constant for the ℓ^2 -norm of the error in the ST-LSPG ROM solution grows only linearly in time, while the stability constant for the ℓ^∞ -norm of the error in the ST-LSPG ROM solution exhibits polynomial growth in time with degree 3/2. Further, this trend is valid for all assessed linear multistep schemes. This highlights one important feature of the proposed method: significantly slower time growth of the solution error in time relative to nonlinear model-reduction methods that perform only spatial projection, as such error bounds grow exponentially in time [38, 28, 34, 11]. This is similar to the slow time growth of the error bounds demonstrated in the context of the space-time reduced-basis method [45, 46, 50, 49].

We now provide computable a posteriori residual-based error bounds and show that the ST-LSPG solution minimizes this bound over all solutions in the space-time trial subspace.

COROLLARY 6.6 (a posteriori error bound). *Under assumptions A1, A2, and A3, the error in the any approximation $\mathbf{w} \in \mathcal{ST}$ can be bounded by the computed residual norm as*

$$\max_{n \in \mathbb{N}(N_t)} \|\mathbf{x}(t^n) - \mathbf{w}(t^n)\|_2 \leq \|\mathbf{x} - \mathbf{w}\|_2 \leq \frac{1}{PK_r} \|\mathbf{r}(\cdot; \mathbf{w})\|_{\bar{\mathbf{A}}^T \bar{\mathbf{A}}}.$$

Further, the ST-LSPG solution is the particular solution for which this error bound is minimized, i.e.,

$$\max_{n \in \mathbb{N}(N_t)} \|\mathbf{x}(t^n) - \tilde{\mathbf{y}}(t^n)\|_2 \leq \|\mathbf{x} - \tilde{\mathbf{y}}\|_2 \leq \frac{1}{PK_r} \min_{\mathbf{w} \in \mathcal{ST}} \|\mathbf{r}(\cdot; \mathbf{w})\|_{\bar{\mathbf{A}}^T \bar{\mathbf{A}}}.$$

Proof. By invoking assumption A3, Lemma 6.2, and norm equivalence $\|\mathbf{w}\|_2 \geq \|\mathbf{w}\|_\infty$, we can derive

$$\|\mathbf{r}(\cdot; \tilde{\mathbf{y}})\|_{\bar{\mathbf{A}}^T \bar{\mathbf{A}}} \geq P \|\mathbf{r}(\cdot; \tilde{\mathbf{y}})\|_2 \geq PK_r \|\mathbf{x} - \tilde{\mathbf{y}}\|_2 \geq PK_r \|\mathbf{x} - \tilde{\mathbf{y}}\|_\infty,$$

which yields the first desired result. The second result follows from applying the optimality property of the ST-LSPG solution (6.1). \square

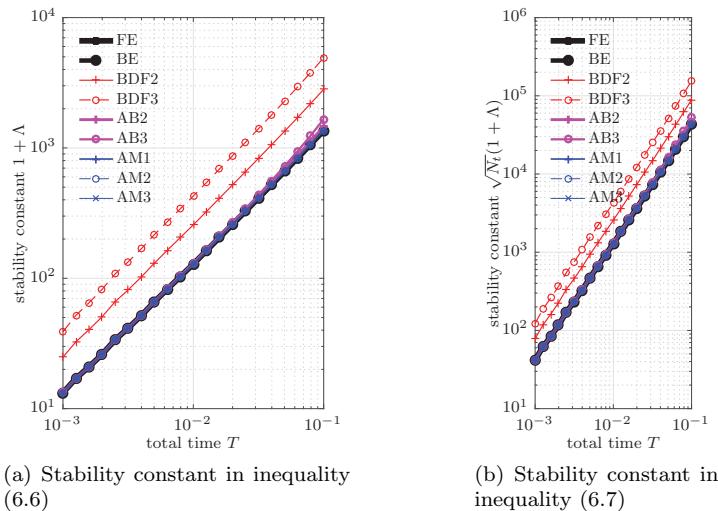


FIG. 1. Stability constants in Corollary 6.5 for time step $\Delta t = 1 \times 10^{-4}$, Lipschitz constant $L_f = 1$, and the following linear multistep methods: backward Euler (BE); backward differentiation formulas (BDF2, BDF3); Adams–Bashforth with $s = 2$ (AB2) and $s = 3$ (AB3); Adams–Moulton with $s = 1$ (AM1), $s = 2$ (AM2), and $s = 3$ (AM3). Note that the stability constant $1 + \Lambda$ for the ℓ^2 -norm error in inequality (6.6) grows linearly in time, while the stability constant $\sqrt{N_t}(1 + \Lambda)$ for the ℓ^∞ -norm error in inequality (6.7) exhibits polynomial time growth with degree 3/2.

7. Numerical experiments. This section compares the performance of the following methods:

- **FOM.** This model corresponds to the full-order model, i.e., the solution satisfying (2.2).
- **LSPG ROM.** This model corresponds to the unweighted LSPG ROM, i.e., the solution that satisfies (3.4) with $\mathbf{A} = \mathbf{I}_{N_s}$.
- **GNAT ROM.** This model corresponds to the GNAT ROM, i.e., the solution that satisfies (3.4) with $\mathbf{A} = (\mathbf{Z}\Phi_r)^+\mathbf{Z}$. Algorithm 5 in [9] is used to construct the sampling matrix \mathbf{Z} .
- **ST-LSPG-1 ROM.** This model corresponds to the unweighted ST-LSPG ROM, i.e., the solution that satisfies (4.4) with $\bar{\mathbf{A}} = \mathbf{I}_{N_s N_t}$. The method is also characterized by the following:
 - Tailored temporal state subspaces computed according to (5.4)–(5.5).
 - As described in section 5.4, interpolation to compute the initial guess. For this, we employ interpolation using linear radial basis functions as described in [14].
- **ST-LSPG-2 ROM.** This model is identical to the ST-LSPG-1 ROM except that it employs fixed temporal subspaces computed according to (5.2)–(5.3).
- **ST-GNAT-1 ROM.** This model corresponds to the ST-GNAT ROM, i.e., the solution that satisfies (4.4) with $\bar{\mathbf{A}} = (\bar{\mathbf{Z}}\bar{\Phi}_r)^+\bar{\mathbf{Z}}$. Otherwise, it is identical to the ST-LSPG-1 ROM with the additional following attributes:
 - Tailored temporal residual subspaces computed according to (5.7)–(5.8).
 - Method 1 in section 5.2 to generate space–time residual samples, where $\mathcal{D}_{\text{res}} = \mathcal{D}_{\text{train}}$.
 - Method 3 in section 5.3 to construct the sampling matrix.
- **ST-GNAT-2 ROM.** This model is identical to the ST-GNAT-1 ROM except

that it employs a fixed temporal state subspace computed according to (5.2)–(5.3).

We assess the accuracy of any ROM solution $\tilde{\mathbf{x}}(\cdot; \boldsymbol{\mu})$ from its mean squared state-space error, i.e.,

$$\text{relative error} = \sqrt{\sum_{n=1}^{N_t} \|\tilde{\mathbf{x}}(t^n; \boldsymbol{\mu}) - \mathbf{x}(t^n; \boldsymbol{\mu})\|_2^2} / \sqrt{\sum_{n=1}^{N_t} \|\mathbf{x}(t^n; \boldsymbol{\mu})\|_2^2},$$

and we measure its computational cost in terms of the wall time incurred by the ROM relative to that incurred by the FOM; the speedup is the reciprocal of the relative wall time. All timings are obtained by performing calculations on an Intel(R) Xeon(R) CPU E5-2670 @ 2.60 GHz, 31.4 GB RAM using the MORTestbed [51] in MATLAB. All reported timings are averaged over five simulations.

7.1. Parameterized Burgers equation. We first consider the parameterized inviscid Burgers equation described in [40], which corresponds to the following initial boundary value problem for $x \in [0, 1]$ and $t \in [0, T]$ with $T = 0.5$:

$$(7.1) \quad \begin{aligned} \frac{\partial w(x, t; \boldsymbol{\mu})}{\partial t} + \frac{\partial f(w(x, t; \boldsymbol{\mu}))}{\partial x} &= 0.02e^{\mu_2 x} \quad \forall x \in [0, 1], \quad \forall t \in [0, T], \\ w(0, t; \boldsymbol{\mu}) &= \mu_1 \quad \forall t \in [0, T], \\ w(x, 0) &= 1 \quad \forall x \in [0, 1], \end{aligned}$$

where $w : [0, 1] \times [0, T] \times \mathcal{D} \rightarrow \mathbb{R}$ is a conserved quantity and the $n_\mu = 2$ parameters comprise the left boundary value and source-term coefficient with $\boldsymbol{\mu} \equiv (\mu_1, \mu_2) \in \mathcal{D} = [1.2, 1.5] \times [0.02, 0.025]$.

After applying Godunov's scheme for spatial discretization with 100 control volumes, (7.1) leads to a parameterized initial-value ODE problem consistent with (2.1) with $N_s = 100$ spatial degrees of freedom. For time discretization, we employ the backward Euler scheme, which is a linear multistep method characterized by $k(t^n) = 1$, $\alpha_0^n = \beta_0^n = 1$, $\alpha_1^n = -1$, $\beta_1^n = 0$, $n \in \mathbb{N}(N_t)$. We employ a uniform time step of $\Delta t = 2.5 \times 10^{-4}$, leading to $N_t = 2000$ time instances. For this problem, all ROMs employ a training set $\mathcal{D}_{\text{train}} = \{1.2, 1.3, 1.4, 1.5\} \times \{0.02, 0.025\}$ such that $n_{\text{train}} = 8$, at which the FOM is solved.

We emphasize that (1) all assessed models employ the same time discretization; this includes the ST-LSPG and ST-GNAT ROMs, as the space-time residual is defined from this time discretization, and (2) we do not consider adaptive time-step selection. Future work will investigate the effect of different time integrators—including those that employ adaptive time-step selection—on the relative performance of the methods.

7.1.1. Space-time bases. Figure 2 plots a selection of spatial and temporal modes computed using the three different techniques proposed in section 5.1. Note that the fixed temporal modes are nearly identical, regardless of whether the T-HOSVD or ST-HOSVD is employed. Thus, because the ST-HOSVD is significantly less computationally expensive, we no longer consider the fixed modes computed with T-HOSVD, which was the approach considered in [6]. On the other hand, the tailored temporal modes are significantly different from the fixed temporal modes. Further, they appear to be well suited for their respective spatial modes, as the temporal bases for higher-index spatial POD modes exhibit higher frequencies, which is consistent with previous studies (e.g., [8]).

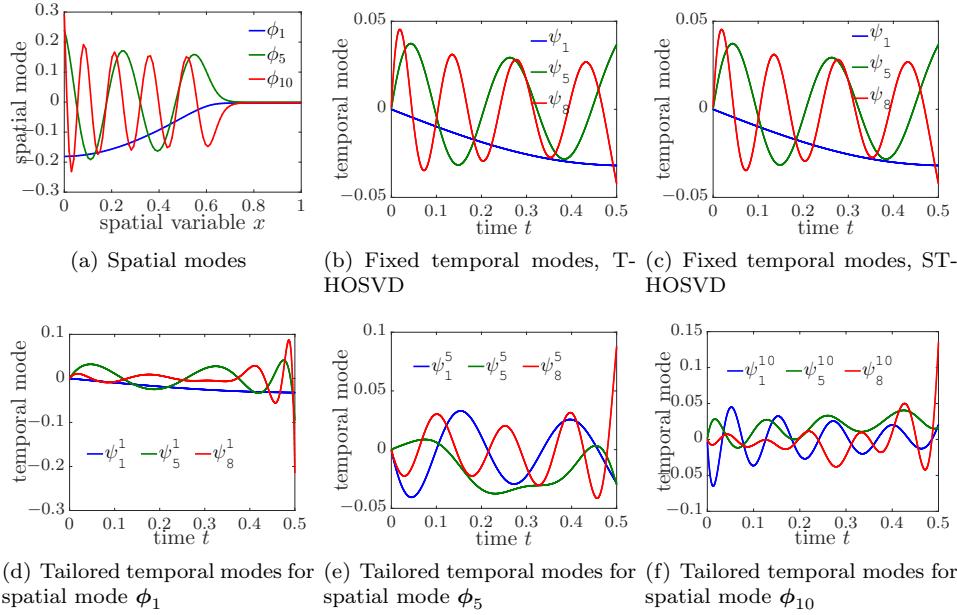


FIG. 2. *Burgers equation. Spatial and temporal modes computed using different tensor-decomposition techniques (see section 5.1).*

7.1.2. Model predictions. We now compare the methods for fixed values of their parameters, and for two randomly selected online points $\mu^1 = (1.35, 0.0229) \notin \mathcal{D}_{\text{train}}$ and $\mu^2 = (1.45, 0.0201) \notin \mathcal{D}_{\text{train}}$. Table 1 reports the method parameter values and the associated performance of the methods. Figure 3 reports snapshots of the methods' responses for $t \in \{0, 0.1665, 0.3332, 0.5\}$.

TABLE 1

Burgers equation. ROM method performance for fixed method parameters at randomly selected online points $\mu^1 = (1.35, 0.0229) \notin \mathcal{D}_{\text{train}}$ and $\mu^2 = (1.45, 0.0201) \notin \mathcal{D}_{\text{train}}$.

Method	LSPG	GNAT	ST-LSPG-1	ST-LSPG-2	ST-GNAT-1	ST-GNAT-2
n_s	15	15	15	15	15	15
n_z		55				
n_r	55					
n_t^i			2		2	
n_t				20		20
\bar{n}_s					30	30
\bar{n}_t					120	120
$n_{r,s}$					100	100
$n_{r,t}^i$					3	10
spatiotemporal dimension	3×10^4	3×10^4	30	300	30	300
relative error for μ^1	0.00074	0.011	0.0025	0.0011	0.0058	0.0063
speedup for μ^1	0.82	0.34	0.34	0.079	7.19	1.78
relative error for μ^2	0.0012	0.017	0.0038	0.0040	0.0077	0.0082
speedup for μ^2	0.80	0.39	0.33	0.080	6.22	1.72

First, note that all ROMs generate accurate responses for this particular combination of parameters, as the relative errors are less than 1% in all cases. Second, note that LSPG generates the most accurate responses but fails to generate any speedup due to its lack of hyper-reduction. GNAT also fails to generate speedup in this case due to the relatively small spatial dimension of the FOM and the larger number of

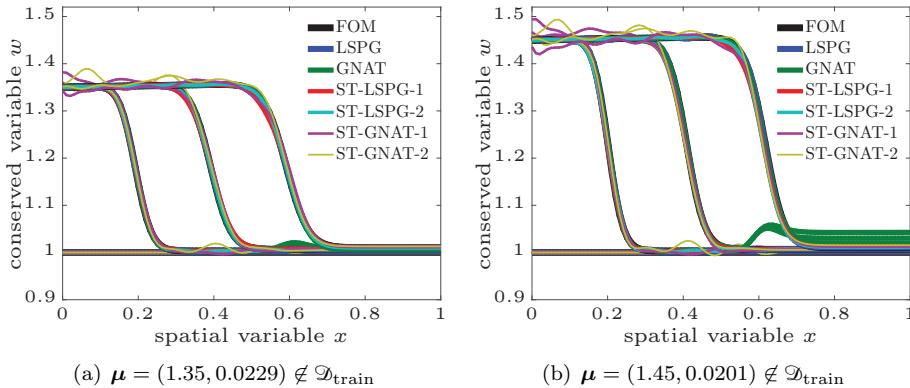


FIG. 3. *Burgers equation. Method solutions for $t \in \{0, 0.1665, 0.3332, 0.5\}$ corresponding to method parameters reported in Table 1.*

Newton iterations required for convergence relative to LSPG. The proposed ST-LSPG methods incur slightly larger errors than the LSPG method, but they do so with orders of magnitude fewer space-time degrees of freedom. This highlights the promise of performing projection in both space and time: the dimensionality of the problem can be significantly reduced while retaining high levels of accuracy. However, due to their lack of hyper-reduction, the ST-LSPG methods do not generate speedups. Finally, by employing hyper-reduction, the ST-GNAT methods generate very accurate predictions with significant speedups. We note that ST-LSPG-1 and ST-GNAT-1 exhibit better overall performance than ST-LSPG-2 and ST-GNAT-2, respectively; this suggests that employing tailored temporal subspaces enables similar accuracy to be achieved using far fewer degrees of freedom, as each temporal basis vector is tailored to its associated spatial basis vector.

7.1.3. Method-parameter study. This section compares the performance of the ROM methods across a variation of all method parameters. This study is essential to objectively compare the methods, as the particular method-parameter values employed in section 7.1.2 did not necessarily yield optimal performance for a given method. For this reason, we subject each model to a parameter study wherein each model parameter is varied between specified limits; Table 2 reports the tested parameter values for each method. We consider all elements in the resulting set if they satisfy the following constraints: $1.5n_s \leq \bar{n}_r \leq n_z$ for GNAT and $1.5n_{st} \leq \bar{n}_r \leq \bar{n}_s \bar{n}_t$ for ST-GNAT-1 and ST-GNAT-2. From these results, we then construct a Pareto front for each method, which is characterized by the method parameters that minimize the competing objectives of relative error and relative wall time.

Figure 4 reports these Pareto fronts for the two online points, as well as an “overall” Pareto front that selects the Pareto-optimal methods across all parameter variations. Table 3 reports values of the method parameters that yielded Pareto-optimal performance. The proposed ST-GNAT-1 method is Pareto-optimal for relative wall times less than one (i.e., faster than the FOM simulation). While the proposed ST-GNAT-2 method does produce speedups, it is dominated by ST-GNAT-1; this provides further evidence of the advantage of employing a tailored relative to a fixed temporal basis. We note that the worst-performing methods correspond to the ST-LSPG-1 and ST-LSPG-2 methods, as their lack of hyper-reduction leads to

TABLE 2

Burgers equation. Parameters used for the method-parameter study. The set of tested parameters comprises the Cartesian product of the specified parameter sets. We consider all elements in the resulting set if they satisfy the following guidelines: $1.5n_s \leq \bar{n}_r \leq n_z$ for GNAT and $1.5n_{st} \leq \bar{n}_r \leq \bar{n}_s \bar{n}_t$ for ST-GNAT-1 and ST-GNAT-2.

Method	LSPG	GNAT	ST-LSPG-1	ST-LSPG-2	ST-GNAT-1	ST-GNAT-2
n_s	$\{10 \times i\}_{i=1}^5$	$\{10 \times i\}_{i=1}^5$	$\{10 \times i\}_{i=1}^5$	$\{10 \times i\}_{i=1}^5$	$\{10 \times i\}_{i=1}^5$	$\{10 \times i\}_{i=1}^5$
n_z		$\{20, 30, 40, 60, 80, 90\}$				
n_r		$\{20, 30, 40, 60, 80, 90\}$				
n_t^i			$\{3, 4, 5, 6, 8\}$		$\{3, 4, 5, 6, 8\}$	
n_t				$\{5, 10, 20, 30\}$		
\bar{n}_s					$\{30, 40\}$	$\{5, 10, 20, 30\}$
\bar{n}_t					$\{60, 120\}$	$\{30, 40, 60, 70, 80\}$
$n_{r,s}^i$					$\{100\}$	$\{120\}$
$n_{r,t}^i$					$\{3\}$	$\{100\}$

significant wall times that far exceed that of the FOM. Further, we note for a fixed error below a certain threshold, the ST-GNAT-1 method is nearly two orders of magnitude faster than the original GNAT method; this can be attributed to the fact that this approach reduces both the spatial and temporal complexities of the FOM. Finally, we note that because the spatial trial subspace employed by LSPG and GNAT has a (relatively large) spatiotemporal dimension of $n_s N_t$, while the space–time trial subspace employed by ST-LSPG and ST-LSPG has a (relatively small) spatiotemporal dimension of $n_{st} (\ll n_s N_t)$, the LSPG and GNAT methods are able to generate smaller errors than the space–time methods. However, this is achieved at significant computational cost that exceeds that of the FOM in this case (i.e., relative wall times greater than one). Thus, for this problem, LSPG is Pareto-optimal and outperforms the space–time ROMs for relative errors less than 10^{-6} , although this regime is not useful because it incurs relative wall times greater than one.

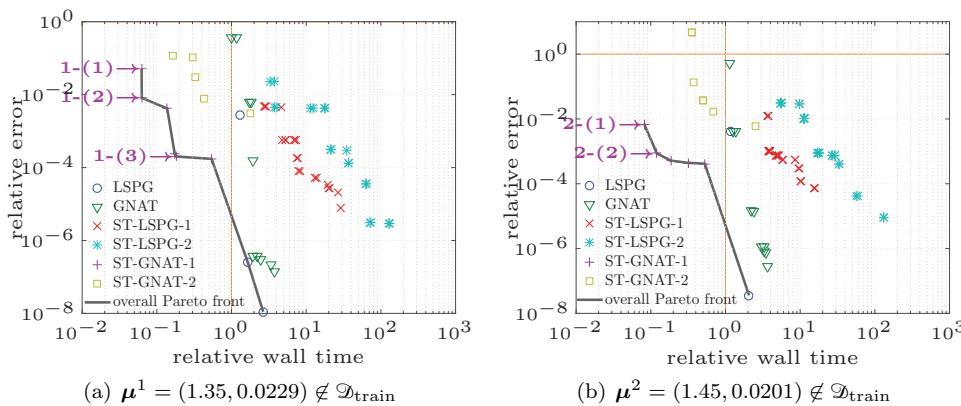


FIG. 4. Burgers equation. Relative error versus relative wall time for varying model parameters reported in Table 1.

7.2. Quasi-1D Euler equation. We now consider a parameterized quasi-1D Euler equation associated with modeling inviscid compressible flow in a one-dimensional converging–diverging nozzle with a continuously varying cross-sectional area [33, Chapter 13]; Figure 5 depicts the problem geometry. The governing system of nonlinear PDEs is

$$\frac{\partial \mathbf{w}}{\partial t} + \frac{1}{A} \frac{\partial(\mathbf{f}(\mathbf{w})A)}{\partial x} = \mathbf{q}(\mathbf{w}) \quad \forall x \in [0, 1] \text{ m}, \quad \forall t \in [0, T],$$

TABLE 3

Burgers equation. Parameter values yielding Pareto-optimal performance for the ST-GNAT-1 method. Figure 4 provides labels.

Label	n_s	n_t^i	\bar{n}_s	\bar{n}_t	$n_{r,s}$	$n_{r,t}^i$
1-(1)	10	3	30	60	100	3
1-(2)	10	4	30	60	100	3
1-(3)	30	3	40	120	100	3
2-(1)	20	3	30	60	100	10
2-(2)	30	3	40	120	100	10

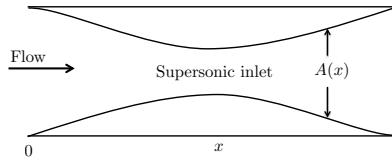


FIG. 5. Quasi-1D Euler. Schematic figures of converging-diverging nozzle.

where $T = 0.6$ s and

$$\mathbf{w} = \begin{bmatrix} \rho \\ \rho u \\ e \end{bmatrix}, \quad \mathbf{f}(\mathbf{w}) = \begin{bmatrix} \rho u \\ \rho u^2 + p \\ (e + p)u \end{bmatrix}, \quad \mathbf{q}(\mathbf{w}) = \begin{bmatrix} 0 \\ \frac{p}{A} \frac{\partial A}{\partial x} \\ 0 \end{bmatrix}, \quad p = (\gamma - 1)\rho e, \quad e = \frac{e}{\rho} - \frac{u^2}{2}, \quad A = A(x).$$

Here, ρ denotes density, u denotes velocity, p denotes pressure, e denotes potential energy per unit mass, ϵ denotes total energy density, γ denotes the specific heat ratio, and A denotes the converging-diverging nozzle cross-sectional area. We employ a specific heat ratio of $\gamma = 1.3$, a specific gas constant of $R = 355.4 \text{ m}^2/\text{s}^2/\text{K}$, a total temperature of $T_t = 300 \text{ K}$, and a total pressure of $p_t = 10^6 \text{ N/m}^2$. The cross-sectional area $A(x)$ is determined by a cubic spline interpolation over the points $(x, A(x)) \in \{(0, 0.2), (0.25, 0.173), (0.5, 0.17), (0.75, 0.173), (1, 0.2)\}$, which results in

$$A(x) = \begin{cases} -0.288x^3 + 0.4080x^2 - 0.1920x + 0.2, & x \in [0, 0.25) \text{ m}, \\ -0.288(x - 0.25)^3 + 0.1920(x - 0.25)^2 - 0.0420(x - 0.25) + 0.1730, & x \in [0.25, 0.5) \text{ m}, \\ 0.288(x - 0.5)^3 - 0.0240(x - 0.5)^2 + 0.17, & x \in [0.5, 0.75) \text{ m}, \\ 0.288(x - 0.75)^3 + 0.1920(x - 0.75)^2 + 0.0420(x - 0.75) + 0.1730, & x \in [0.75, 1] \text{ m}. \end{cases}$$

We assume a perfect gas that obeys the ideal gas law (i.e., $p = \rho RT$). The initial flow field is created in several steps. First, the following isentropic relations are used to generate a zero pressure-gradient flow field:

$$M(x) = \frac{M_m A_m}{A(x)} \left(\frac{1 + \frac{\gamma-1}{2} M(x)^2}{1 + \frac{\gamma-1}{2} M_m^2} \right)^{\frac{\gamma+1}{2(\gamma-1)}}, \quad p(x) = p_t \left(1 + \frac{\gamma-1}{2} M(x)^2 \right)^{\frac{-\gamma}{\gamma-1}},$$

$$T(x) = T_t \left(1 + \frac{\gamma-1}{2} M(x)^2 \right)^{-1}, \quad \rho(x) = \frac{p(x)}{RT(x)}, \quad c(x) = \sqrt{\frac{p(x)}{\rho(x)}}, \quad u(x) = M(x)c(x),$$

where a subscript m indicates the flow quantity at $x = 0.5$ m, and M denotes the Mach number. Then, a shock is placed at $x = 0.85$ m of the flow field. We use the jump relations for a stationary shock and the perfect gas equation of state to derive the velocity across the shock u_2 , which satisfies the quadratic equation

$$(7.2) \quad \left(\frac{1}{2} - \frac{\gamma}{\gamma-1} \right) u_2^2 + \frac{\gamma}{\gamma-1} \frac{n}{m} u_2 - h = 0.$$

Here, $m := \rho_2 u_2 = \rho_1 u_1$, $n := \rho_2 u_2^2 + p_2 = \rho_1 u_1^2 + p_1$, $h := (e_2 + p_2)/\rho_2 = (e_1 + p_1)/\rho_1$, and subscripts 1 and 2 denote a flow quantity to the left and to the right of the shock, respectively. We employ the solution u_2 to (7.2), which leads to a discontinuity (i.e., shock). Finally, the exit pressure is increased to a factor P_{exit} of its original value in order to generate transient dynamics.

Applying a finite-volume spatial discretization with 50 equally spaced control volumes and fully implicit boundary conditions leads to a parameterized system of nonlinear ODEs consistent with (2.1) with $N_s = 150$ spatial degrees of freedom. The Roe flux difference vector splitting method is used to compute the flux at each intercell face [33, Chapter 9]. For time discretization, we again apply the backward Euler scheme and a uniform time step of $\Delta t = 0.001$ s, leading to $N_t = 600$.

For this problem, we use the following two parameters: the pressure factor $\mu_1 = P_{\text{exit}}$ and the Mach number at the middle of the nozzle $\mu_2 = M_m$. All ROMs employ a training set, at which the FOM is solved, of $\mathcal{D}_{\text{train}} = \{1.7 + 0.01i\}_{i=0}^3 \times \{1.7, 1.72\}$ such that $n_{\text{train}} = 8$.

7.2.1. Space-time bases. Figure 6 plots several spatial and temporal modes computed using the three different techniques proposed in section 5.1. As with the Burgers equation, the “fixed” temporal modes are nearly identical, regardless of whether the T-HOSVD or ST-HOSVD is employed, rendering the ST-HOSVD more appealing due to its reduced computational cost. In addition, the tailored temporal modes are significantly different, with the temporal basis exhibiting higher frequencies for higher-index spatial modes, as expected.

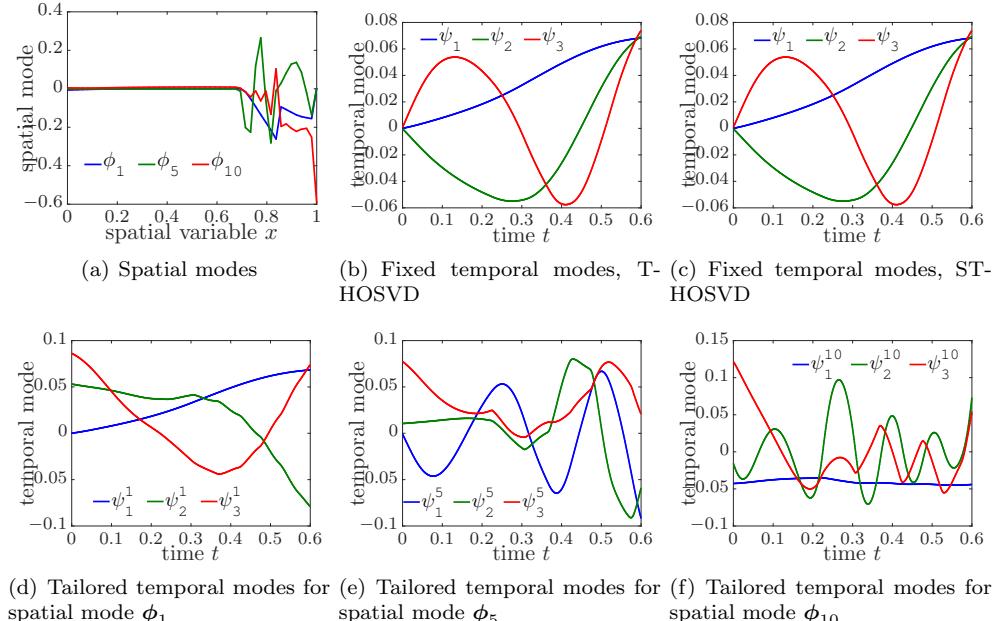


FIG. 6. Quasi-1D Euler equation. Spatial and temporal modes computed using different techniques (see section 5.1).

7.2.2. Model predictions. We now compare the methods for fixed values of their parameters, and for two randomly selected online points $\mu^1 = (1.7125, 1.71) \notin \mathcal{D}_{\text{train}}$ and $\mu^2 = (1.7225, 1.705) \notin \mathcal{D}_{\text{train}}$. Table 4 reports the method parameter

values and the associated performance of the methods. Figure 7 reports snapshots of the methods' responses for $t \in \{0, T\}$.

TABLE 4

Quasi-1D Euler equation. ROM method performance for fixed method parameters at randomly selected online points $\mu^1 = (1.7125, 1.71) \notin \mathcal{D}_{train}$ and $\mu^2 = (1.7225, 1.705) \notin \mathcal{D}_{train}$.

Method	LSPG	GNAT	ST-LSPG-1	ST-LSPG-2	ST-GNAT-1	ST-GNAT-2
n_s	50	50	50	50	50	50
n_z		145				
n_r		145				
n_t^i			3		3	
n_t				30		30
\tilde{n}_s					120	140
\tilde{n}_t					20	100
$n_{r,s}$					150	150
$n_{r,t}^i$					10	10
spatiotemporal dimension	3×10^4	3×10^4	150	1.5×10^3	150	1.5×10^3
relative error for μ^1	7.78×10^{-6}	0.55	0.012	6.3×10^{-4}	0.0023	0.0048
speedup for μ^1	0.77	1.04	0.84	0.58	21.79	0.49
relative error for μ^2	8.31×10^{-6}	0.026	0.0076	0.0021	0.0025	0.0040
speedup for μ^2	0.81	1.01	0.85	0.41	22.52	2.79

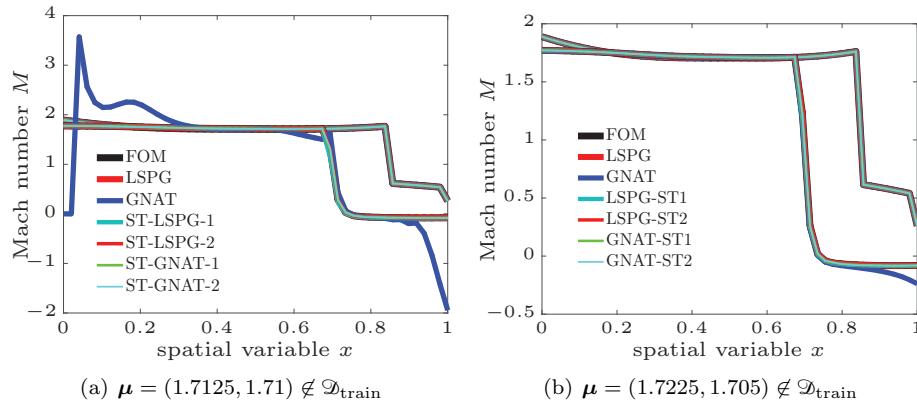


FIG. 7. *Quasi-1D Euler equation. Method solutions for $t \in \{0, T\}$ corresponding to method parameters reported in Table 4.*

Conclusions are similar to those derived from the Burgers equation results. First, note that all ROMs except for GNAT in the case of μ^1 generate accurate responses, as the relative errors are less than 3% in all cases. Second, as before, LSPG generates the most accurate responses but fails to generate any speedup due to its lack of hyper-reduction. The proposed ST-LSPG methods incur sub-2% errors, but they do so with orders of magnitude fewer spatiotemporal degrees of freedom relative to the LSPG and GNAT methods, which highlights the promise of performing projection in both space and time. Again, as these methods do not employ hyper-reduction, they do not generate speedups. Finally, the ST-GNAT methods generate both accurate predictions with significant speedups. We note that ST-GNAT-1 performs better than ST-GNAT-2, providing further evidence of the ability of tailored bases to produce accurate responses with fewer degrees of freedom.

7.2.3. Method-parameter study. We again compare the performance of the ROM methods across a wide variation of all method parameters. Table 5 reports the tested parameter values for each method. We consider all elements in the resulting

TABLE 5

Quasi-1D Euler equation. Parameters used for the method-parameter study. The set of tested parameters comprises the Cartesian product of the specified parameter sets. We consider all elements in the resulting set if they satisfy the following guidelines: $1.5n_s \leq \bar{n}_r \leq n_z$ for GNAT and $1.5n_{st} \leq \bar{n}_r \leq \bar{n}_s \bar{n}_t$ for ST-GNAT-1 and ST-GNAT-2.

Method	LSPG	GNAT	ST-LSPG-1	ST-LSPG-2	ST-GNAT-1	ST-GNAT-2
n_s	$\{10 \times i\}_{i=1}^6$	$\{10 \times i\}_{i=1}^6$	$\{10 \times i\}_{i=1}^6$	$\{10 \times i\}_{i=1}^6$	$\{10 \times i\}_{i=1}^6$	$\{10 \times i\}_{i=1}^6$
n_z		$\{10 \times i\}_{i=2}^{10} \cup \{120, 145\}$				
n_r		$\{10 \times i\}_{i=2}^{10} \cup \{120, 145\}$				
n_t^i			$\{i\}_{i=3}^8$	$\{10 \times i\}_{i=2}^5$	$\{i\}_{i=3}^8$	$\{10 \times i\}_{i=2}^5$
n_t					$\{120\}$	$\{140\}$
\bar{n}_s					$\{10, 20, 30, 40, 60\}$	$\{30, 50, 100, 150\}$
\bar{n}_t					150	150
$n_{r,s}$					10	10
$n_{r,t}^i$						

set if they satisfy constraints $1.5n_s \leq \bar{n}_r \leq n_z$ for GNAT and $1.5n_{st} \leq \bar{n}_r \leq \bar{n}_s \bar{n}_t$ for ST-GNAT-1 and ST-GNAT-2. From these results, we then construct a Pareto front for each method, which is characterized by the method parameters that minimize the competing objectives of relative error and relative wall time.

Figure 8 reports these Pareto fronts for the two online points, as well as an overall Pareto front that selects the Pareto-optimal methods across all parameter variations. Table 6 reports values of the method parameters that yielded Pareto-optimal performance. These results show that—as before—the proposed ST-GNAT-1

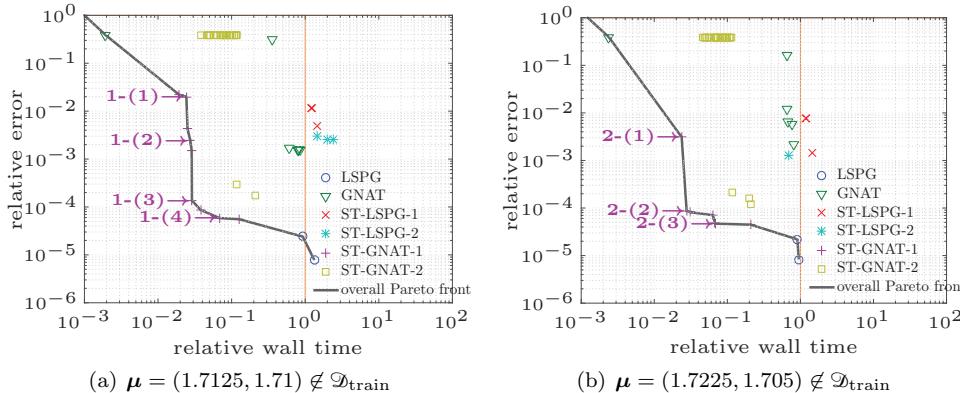


FIG. 8. *Quasi-1D Euler equation. Relative error versus relative wall time for varying model parameters reported in Table 5.*

TABLE 6

Quasi-1D Euler equation. Parameter values yielding Pareto-optimal performance for the ST-GNAT-1 method. Figure 8 provides labels.

Label	n_s	n_t^i	\bar{n}_s	\bar{n}_t	$n_{r,s}$	$n_{r,t}^i$
1-(1)	10	3	120	10	150	10
1-(2)	60	3	120	10	150	10
1-(3)	30	4	120	10	150	10
1-(4)	50	8	120	10	150	10
2-(1)	30	3	120	10	150	10
2-(2)	50	4	120	10	150	10
2-(3)	60	7	120	10	150	10

method is Pareto optimal for relative wall time less than 0.9 and relative errors less than 20%. While the proposed ST-GNAT-2 method produces speedups, it is again dominated by ST-GNAT-1, further highlighting the advantage of tailored versus fixed temporal bases. Again, the worst-performing methods correspond to the ST-LSPG-1 and ST-LSPG-2 methods, as their lack of hyper-reduction leads to significant wall times that far exceed that of the FOM. Further, we note that for a fixed error below a certain threshold, the ST-GNAT-1 method is over one order of magnitude faster than the original GNAT method; this can be attributed to the fact that this approach reduces both the spatial and temporal complexities of the FOM. Finally, we again note that LSPG is Pareto-optimal and outperforms the space-time ROMs for extremely small relative errors less than approximately 3×10^{-5} due to the higher spatiotemporal dimensionality of the spatial trial subspace; however, this regime is not useful for this problem, as it leads to LSPG models roughly as expensive as the FOM (i.e., relative wall times near one).

8. Conclusions. This work proposed a model-reduction method for nonlinear dynamical systems based on space-time least-squares Petrov–Galerkin projection. The method computes optimal approximations by minimizing the discrete space–time residual over all elements in a low-dimensional space–time trial subspace in a weighted ℓ^2 -norm. Advantages of the method include

- its ability to reduce both the spatial and temporal dimensions of the dynamical system (Remark 4.1),
- a priori error bounds that bound the solution error by the best space–time approximation error and whose stability constants exhibit subquadratic growth in time (section 6),
- applicability to general nonlinear dynamical systems,
- hyper-reduction that reduces the complexity in the presence of general nonlinearities (section 4.3), and
- its ability to extract multiple space–time basis vectors from each training simulation via tensor decomposition (section 6).

In addition to introducing the novel ST-LSPG method, this work proposed specific approaches to computing the method’s ingredients: the space–time trial subspace (section 5.1), the space–time residual basis in the case of ST–GNAT (section 5.2), the sampling matrix in the case of hyper-reduction (section 5.3), and the initial guess used in the Gauss–Newton method applied to solve the nonlinear least-squares problem (section 5.4). Numerical experiments demonstrated the ability of the proposed method to generate orders-of-magnitude speedups over existing spatial-projection-based ROMs without sacrificing accuracy.

Future work entails implementing the method in parallel computational-mechanics codes, devising techniques to reduce the amount of storage required for the state and residual tensors, and assessing the effect of different time integrators—as well as adaptive time-stepping—on method performance.

Acknowledgments. The authors gratefully acknowledge Tamara Kolda and Grey Ballard for insightful discussions that led to the tensor-decomposition approaches to computing the space–time trial subspace. The authors also acknowledge Professors Benjamin Peherstorfer and Masayuki Yano for useful comments received at the Model Reduction for Parametrized Systems (MoRePaS) III workshop and the 2017 SIAM Conference on Computational Science and Engineering, respectively. The authors also gratefully acknowledge the helpful comments provided by the anonymous reviewers.

REFERENCES

- [1] P. ASTRID, S. WEILAND, K. WILLCOX, AND T. BACKX, *Missing point estimation in models described by proper orthogonal decomposition*, IEEE Trans. Automat. Control, 53 (2008), pp. 2237–2251.
- [2] W. AUSTIN, G. BALLARD, AND T. G. KOLDA, *Parallel Tensor Compression for Large-Scale Scientific Data*, preprint, <https://arxiv.org/abs/1510.06689>, 2015.
- [3] I. BABUŠKA, R. TEMPONE, AND G. E. ZOURARIS, *Galerkin finite element approximations of stochastic elliptic partial differential equations*, SIAM J. Numer. Anal., 42 (2004), pp. 800–825, <https://doi.org/10.1137/S0036142902418680>.
- [4] A. T. BARKER, T. REES, AND M. STOLL, *A fast solver for an \mathcal{H}_1 regularized PDE-constrained optimization problem*, Commun. Comput. Phys., 19 (2016), pp. 143–167.
- [5] M. BARRAULT, Y. MADAY, N. C. NGUYEN, AND A. T. PATERA, *An ‘empirical interpolation’ method: Application to efficient reduced-basis discretization of partial differential equations*, C. R. Math. Acad. Sci. Paris, 339 (2004), pp. 667–672.
- [6] M. BAUMANN, P. BENNER, AND J. HEILAND, *Space-Time Galerkin POD with Application in Optimal Control of Semi-linear Parabolic Partial Differential Equations*, preprint, <https://arxiv.org/abs/1611.04050>, 2016.
- [7] R. BOS, X. BOMBOIS, AND P. VAN DEN HOF, *Accelerating large-scale non-linear models for monitoring and control using spatial and temporal correlations*, in Proceedings of the 2004 American Control Conference (Boston, MA), IEEE, Washington, DC, 2004, pp. 3705–3710.
- [8] K. CARLBERG, M. BARONE, AND H. ANTIL, *Galerkin v. least-squares Petrov–Galerkin projection in nonlinear model reduction*, J. Comput. Phys., 330 (2017), pp. 693–734.
- [9] K. CARLBERG, C. BOU-MOSLEH, AND C. FARHAT, *Efficient non-linear model reduction via a least-squares Petrov–Galerkin projection and compressive tensor approximations*, Internat. J. Numer. Methods Engrg., 86 (2011), pp. 155–181.
- [10] K. CARLBERG, L. BRENCHER, B. HAASDONK, AND A. BARTH, *Data-Driven Time Parallelism via Forecasting*, preprint, <https://arxiv.org/abs/1610.09049>, 2016.
- [11] 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*, J. Comput. Phys., 242 (2013), pp. 623–647.
- [12] K. CARLBERG, J. RAY, AND B. VAN BLOEMEN WAANDERS, *Decreasing the temporal complexity for nonlinear, implicit reduced-order models by forecasting*, Comput. Methods Appl. Mech. Engrg., 289 (2015), pp. 79–103.
- [13] S. CHATURANTABUT AND D. C. SORENSEN, *Nonlinear model reduction via discrete empirical interpolation*, SIAM J. Sci. Comput., 32 (2010), pp. 2737–2764, <https://doi.org/10.1137/090766498>.
- [14] A. CHIROKOV, *Scattered data interpolation and approximation using radial base functions*, MATLAB Central Exchange (retrieved Feburary 18, 2006).
- [15] Y. CHOI, *Simultaneous Analysis and Design in PDE-Constrained Optimization*, Ph.D. thesis, Stanford University, Stanford, CA, 2012.
- [16] Y. CHOI, C. FARHAT, W. MURRAY, AND M. SAUNDERS, *A practical factorization of a Schur complement for PDE-constrained distributed optimal control*, J. Sci. Comput., 65 (2015), pp. 576–597.
- [17] P. G. CONSTANTINE AND Q. WANG, *Residual minimizing model interpolation for parameterized nonlinear dynamical systems*, SIAM J. Sci. Comput., 34 (2012), pp. A2118–A2144, <https://doi.org/10.1137/100816717>.
- [18] L. DE LATHAUWER, B. DE MOOR, AND J. VANDEWALLE, *A multilinear singular value decomposition*, SIAM J. Matrix Anal. Appl., 21 (2000), pp. 1253–1278, <https://doi.org/10.1137/S0895479896305696>.
- [19] M. DROHMANN, B. HAASDONK, AND M. OHLBERGER, *Reduced basis approximation for nonlinear parametrized evolution equations based on empirical operator interpolation*, SIAM J. Sci. Comput., 34 (2012), pp. A937–A969, <https://doi.org/10.1137/10081157X>.
- [20] R. ELBER, *Perspective: Computer simulations of long time dynamics*, J. Chem. Phys., 144 (2016), 060901.
- [21] R. EVERSON AND L. SIROVICH, *Karhunen–Loève procedure for gappy data*, J. Opt. Soc. Amer. A, 12 (1995), pp. 1657–1664.
- [22] R. D. FALGOUT, S. FREIDHOFF, T. V. KOLEV, S. P. MACLACHLAN, AND J. B. SCHRODER, *Parallel time integration with multigrid*, SIAM J. Sci. Comput., 36 (2014), pp. C635–C661, <https://doi.org/10.1137/130944230>.

- [23] C. FARHAT AND M. CHANDESRIS, *Time-decomposed parallel time-integrators: Theory and feasibility studies for fluid, structure, and fluid-structure applications*, Internat. J. Numer. Methods Engrg., 58 (2003), pp. 1397–1434.
- [24] D. GALBALLY, K. FIDKOWSKI, K. WILLCOX, AND O. GHATTAS, *Non-linear model reduction for uncertainty quantification in large-scale inverse problems*, Internat. J. Numer. Methods Engrg., 81 (2009), pp. 1581–1608.
- [25] R. G. GHANEM AND P. D. SPANOS, *Stochastic Finite Elements: A Spectral Approach*, revised ed., Dover, Mineola, NY, 2003.
- [26] R. HAFTKA AND M. KAMAT, *Simultaneous nonlinear structural analysis and design*, Comput. Mech., 4 (1989), pp. 409–416.
- [27] T. ITO AND K. TANIKAWA, *Long-term integrations and stability of planetary orbits in our solar system*, Monthly Notices Roy. Astronom. Soc., 336 (2002), pp. 483–500.
- [28] D. J. KNEZEVIC, N.-C. NGUYEN, AND A. T. PATERA, *Reduced basis approximation and a posteriori error estimation for the parametrized unsteady Boussinesq equations*, Math. Models Methods Appl. Sci., 21 (2011), pp. 1415–1442.
- [29] P. KRYSL, S. LALL, AND J. E. MARSDEN, *Dimensional model reduction in non-linear finite elements dynamics of solids and structures*, Internat. J. Numer. Methods Engrg., 51 (2001), pp. 479–504.
- [30] P. A. LEGRESLEY, *Application of Proper Orthogonal Decomposition (POD) to Design Decomposition Methods*, Ph.D. thesis, Stanford University, Stanford, CA, 2006.
- [31] J. LIONS, Y. MADAY, AND G. TURINICI, *A “parareal” in time discretization of PDEs*, C. R. Acad. Sci. Paris Sér. I Math., 332 (2001), pp. 661–668 (in French).
- [32] D. J. LUCIA, P. S. BERAN, AND W. A. SILVA, *Reduced-order modeling: New approaches for computational physics*, Progr. Aerospace Sci., 40 (2004), pp. 51–117.
- [33] R. MACCORMACK, *Numerical Computation of Compressible Viscous Flow*, Technical report, Lecture notes for AA214b and AA214c, Stanford University, Stanford, CA, 2007.
- [34] N.-C. NGUYEN, G. ROZZA, AND A. T. PATERA, *Reduced basis approximation and a posteriori error estimation for the time-dependent viscous Burgers’ equation*, Calcolo, 46 (2009), pp. 157–185.
- [35] J. NOCEDAL AND S. J. WRIGHT, *Numerical Optimization*, 2nd ed., Springer, New York, 2006.
- [36] C. E. OROZCO AND O. N. GHATTAS, *A reduced SAND method for optimal design of non-linear structures*, Internat. J. Numer. Methods Engrg., 40 (1997), pp. 2759–2774.
- [37] J. W. PEARSON AND A. J. WATHEN, *A new approximation of the Schur complement in preconditioners for PDE-constrained optimization*, Numer. Linear Algebra Appl., 19 (2012), pp. 816–829.
- [38] M. RATHINAM AND L. R. PETZOLD, *A new look at proper orthogonal decomposition*, SIAM J. Numer. Anal., 41 (2003), pp. 1893–1925, <https://doi.org/10.1137/S0036142901389049>.
- [39] T. REES, M. STOLL, AND A. WATHEN, *All-at-once preconditioning in PDE-constrained optimization*, Kybernetika, 46 (2010), pp. 341–360.
- [40] M. J. REWIENSKI, *A Trajectory Piecewise-Linear Approach to Model Order Reduction of Non-linear Dynamical Systems*, Ph.D. thesis, Massachusetts Institute of Technology, Cambridge, MA, 2003.
- [41] D. RYCKELYNCK, *A priori hyperreduction method: An adaptive approach*, J. Comput. Phys., 202 (2005), pp. 346–366.
- [42] B. SPACE, H. RABITZ, AND A. ASKAR, *Long time scale molecular dynamics subspace integration method applied to anharmonic crystals and glasses*, J. Chem. Phys., 99 (1993), pp. 9070–9079.
- [43] Z. A. TAYLOR, S. CROZIER, AND S. OURSELIN, *Real-time surgical simulation using reduced order finite element analysis*, in Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention, Springer, New York, 2010, pp. 388–395.
- [44] L. R. TUCKER, *Some mathematical notes on three-mode factor analysis*, Psychometrika, 31 (1966), pp. 279–311.
- [45] K. URBAN AND A. PATERA, *A new error bound for reduced basis approximation of parabolic partial differential equations*, C. R. Math. Acad. Sci. Paris, 350 (2012), pp. 203–207.
- [46] K. URBAN AND A. PATERA, *An improved error bound for reduced basis approximation of linear parabolic problems*, Math. Comp., 83 (2014), pp. 1599–1615.
- [47] N. VANNIEUWENHOVEN, R. VANDEBRIL, AND K. MEERBERGEN, *A new truncation strategy for the higher-order singular value decomposition*, SIAM J. Sci. Comput., 34 (2012), pp. A1027–A1052, <https://doi.org/10.1137/110836067>.
- [48] S. VOLKWEIN AND S. WEILAND, *An algorithm for Galerkin projections in both time and spatial coordinates*, in Proceedings of the 17th International Symposium on Mathematical Theory of Networks and Systems (MTNS), Kyoto, Japan, 2006, ThP03.1.

- [49] M. YANO, *A space-time Petrov–Galerkin certified reduced basis method: Application to the Boussinesq equations*, SIAM J. Sci. Comput., 36 (2014), pp. A232–A266, <https://doi.org/10.1137/120903300>.
- [50] M. YANO, A. T. PATERA, AND K. URBAN, *A space-time hp-interpolation-based certified reduced basis method for Burgers' equation*, Math. Models Methods Appl. Sci., 24 (2014), pp. 1903–1935.
- [51] M. J. ZAHR, K. CARLBERG, D. AMSALLEM, AND C. FARHAT, *Comparison of Model Reduction Techniques on High-Fidelity Linear and Nonlinear Electrical, Mechanical, and Biological Systems*, Technical report, University of California, Berkeley, CA, 2010.