

Block-structured Operator Inference for coupled multiphysics model reduction

Benjamin G. Zastrow ^{*}, Anirban Chaudhuri [†], and Karen E. Willcox [‡]
University of Texas at Austin, Austin, TX, 78712

Anthony Ashley [§] and Michael Chamberlain Henson [¶]
Lockheed Martin, Fort Worth, TX, 76101

This paper presents a block-structured formulation of Operator Inference as a way to learn structured reduced-order models for multiphysics systems. The approach specifies the governing equation structure for each physics component and the structure of the coupling terms. Once the multiphysics structure is specified, the reduced-order model is learned from snapshot data following the nonintrusive Operator Inference methodology. In addition to preserving physical system structure, which in turn permits preservation of system properties such as stability and second-order structure, the block-structured approach has the advantages of reducing the overall dimensionality of the learning problem and admitting tailored regularization for each physics component. The numerical advantages of the block-structured formulation over a monolithic Operator Inference formulation are demonstrated for aeroelastic analysis, which couples aerodynamic and structural models. For the benchmark test case of the AGARD 445.6 wing, block-structured Operator Inference provides an average 20% online prediction speedup over monolithic Operator Inference across subsonic and supersonic flow conditions in both the stable and fluttering parameter regimes while preserving the accuracy achieved with monolithic Operator Inference.

I. Introduction

Building reduced-order models (ROMs) for coupled multiphysics systems requires addressing additional challenges beyond those posed by a single-physics model reduction problem. For example, we often wish to preserve certain features and properties of a single physics regime’s physical or numerical model, but model reduction methods that view the system monolithically tend to ignore this type of information. In this paper, we show how block-structured Operator Inference exploits knowledge of the block structure of coupled multiphysics systems to improve the ROM

Presented as Paper 2023-0330 at the 2023 AIAA SciTech Forum, National Harbor, MD, January 23-27, 2023.

^{*}Graduate Research Assistant, Department of Aerospace Engineering and Engineering Mechanics, bzastrow@utexas.edu, AIAA Student Member.

[†]Research Scientist, Oden Institute for Computational Engineering and Sciences, anirbanc@oden.utexas.edu, AIAA Member.

[‡]Professor, Department of Aerospace Engineering and Engineering Mechanics, kwillcox@oden.utexas.edu, AIAA Fellow.

[§]Aeronautical Engineer Staff, CFD, anthony.s.ashley@lmco.com, AIAA Member.

[¶]Fellow, Transformational Solutions/AeroIT, mike.c.henson@lmco.com, AIAA Associate Fellow.

while preserving the advantages of standard, monolithic Operator Inference.

Model reduction aims to represent high-dimensional dynamics with a lower-dimensional system while maintaining a sufficient level of predictive accuracy. The ROMs can be run at orders of magnitude lower computational cost compared to the full-order models. Projection-based approaches like the proper orthogonal decomposition (POD) [1–4] use the trajectory data from a full-order model to derive a reduced basis, then the full-order operators of the governing equations are projected onto the reduced basis and the dynamics are integrated forward in time [5, 6]. Another technique, dynamic mode decomposition (DMD) [7, 8], uses the full-order trajectory data to construct a linear reduced operator to approximate the system, thus enabling analysis of eigenmodes and eigenvalues that often can reasonably characterize even nonlinear full-order dynamics. However, in the case of multiphysics systems such as aeroelasticity, these types of model reduction methods are often too general to fully embed the knowledge of the nature of the coupling between systems into the ROM, and thus gains in accuracy, robustness, and speed can be missed.

Coupled model reduction methods directly incorporate knowledge of the multiphysics setting into the ROM construction process. For example, it is common practice in structural dynamics to use component mode synthesis via the Hurty/Craig-Bampton method to break up a system of interacting structural components into separate submodels that only interact along specific sets of boundary degrees of freedom [9, 10]. A review of model reduction methods for coupled systems that focuses on linear, time-invariant systems represented via transfer functions is given in [11], focusing on application of balanced truncation and moment matching methods. Another review of coupled model reduction methods in [12] separates different approaches based on whether the dynamics are coupled through internal states or through inputs and outputs. In the nonintrusive setting, the authors in [13] learn boundary response maps from data to couple models without requiring direct access to the governing equations. Another nonintrusive approach exploits knowledge of the grid adjacency of a fluid-structure interaction problem to efficiently infer a sparse full-order model, after which POD is used to project the operators to a reduced subspace to construct a ROM [14]. From the block-structured perspective, the inherent structure of the governing equations that arises due to coupling is identified in the full-order model structure and then deliberately preserved during construction of the ROM.

Methods for structure-preserving model reduction and coupled model reduction often overlap due to their many shared characteristics. For example, it is common for the governing ODEs of a coupled ROM to possess a particular block structure that can be preserved through the design of a targeted model reduction approach [11, 12, 15–17]. Block structure preservation can be seen as a type of physics-informed modeling because the block structure directly represents the form of each single-physics governing equation and also represents the form of the coupling between the physics regimes. In this paper, we use another physics-informed method, Operator Inference, to replace the intrusive Galerkin projection step with a nonintrusive linear least squares problem which still targets a ROM form dictated by the full-order model’s physics, thus combining the benefits of data-driven learning and physics-informed modeling [18, 19]. Another form of structure preservation for the special case of systems governed by nonlinear Lagrangian dynamics appears in [20],

where the authors separate the system into linear components learned via Operator Inference and nonlinear components learned via polynomial-augmented multilayer perceptrons, thus utilizing the power of structure-preserving ROMs to embed knowledge of the nonlinear Lagrangian setting into the model. In [21], the block structure of the governing PDEs for a batch chromatography system is preserved in a similar manner to this paper by creating a block-structured projection matrix with each block corresponding to the basis for a single subsystem. The authors then infer the linear operators and intrusively impose the known nonlinear terms.

In this paper, we propose a block-structured, coupled multiphysics Operator Inference ROM obtained via a combination of imposition of known structure and nonintrusive inference of unknown operators. We formulate this approach in the context of a high-dimensional aeroelastic problem where we do not assume the nonlinear terms to be known, and therefore we infer the nonlinear operators nonintrusively as well, which also requires the inclusion of regularization in the inference subproblems. Incorporating knowledge of a system’s block structure into Operator Inference permits greater flexibility than standard, monolithic Operator Inference and enables coupling of multiple physics regimes into a single, multiphysics ROM, while preserving the benefits of Operator Inference’s nonintrusive approach. We leverage the added modularity of the block-structured operators to specify the governing equation structure of each physics regime separately, which reduces computational expense by reducing the number of operator terms to be learned.

We demonstrate our approach using the Advisory Group for Aerospace Research and Development (AGARD) 445.6 Wing [22] as a high-dimensional coupled multiphysics example application in the aeroelastic setting. The AGARD wing has been studied frequently in the model reduction context in the literature. In [23], the author uses Walsh functions to simultaneously excite multiple impulse responses in a computational fluid dynamics (CFD) model. The eigensystem realization algorithm (ERA) [24] and nonlinear aeroelastic system identification via Volterra theory [25] are then used to convert these impulses into an unsteady aerodynamic ROM that can be coupled to a structural model to create a coupled aeroelastic ROM [26, 27]. This Walsh function technique also is used in the POD methodology in [28], where the authors build the reduced basis via an incremental approach to avoid handling the full snapshot matrix all at once. In this work, we take a somewhat different approach and build nonintrusive ROMs for coupled multi-physics problems like the AGARD 445.6 wing via our block-structured Operator Inference method. Reduced-order modeling has long been identified as an effective way to reduce the cost of aeroelastic analyses and flutter boundary calculations. Past seminal work has shown the effectiveness of POD-based ROMs for fluid-structure interaction [29, 30] and aeroelastic assessments of full aircraft configurations [31–33], vortex lattice models with structural modal decompositions for flutter prediction [34], Kriging partial least squares models for aerodynamic design [35], and the frequency-domain Karhunen-Loeve method for fluid dynamics ROMs [36]. In this paper, our target is to achieve similar computational cost reductions for aeroelastic analysis, but to do so with nonintrusive methodology, meaning that our approach can be easily applied to legacy and commercial high-fidelity solvers where the user may not have access to the code at the level

required for intrusive model reduction approaches.

We generate high-dimensional simulation data using NASA’s FUN3D software [37], where the full-order model consists of a fluid dynamics finite volume model coupled with a structural dynamics finite element modal decomposition model. FUN3D’s aeroelasticity capability couples the structural and fluid dynamics and generates high-fidelity snapshot training data for the AGARD wing. The main contributions of this paper are:

- (i) *A formulation and algorithm for block-structured Operator Inference with regularization:* We exploit the block structure of the governing equations to specify distinct structure and separately tailor the regularization for each regime of the dynamical system.
- (ii) *Computational speedups due to block-structuring.* Block-structured Operator Inference provides reduced computational complexity during both inference and prediction. This reduced complexity leads to lower computational costs that improve the efficiency of Operator Inference ROMs.
- (iii) *Application of block-structured Operator Inference to multiphysics problems.* Block-structured Operator Inference adapts the Operator Inference method to the multiphysics setting by embedding knowledge of the coupling between physics regimes into the learning problem.

The remainder of this paper is organized as follows. Section II describes the Operator Inference method, first as initially presented in [18] and then with the block-structured formulation that is the main contribution of this paper. Section III presents the AGARD 445.6 wing aeroelastic modeling problem. Section IV compares the performance of the block-structured and monolithic Operator Inference methods in the context of the AGARD wing. Section V provides concluding remarks.

II. Block-structured Operator Inference for multiphysics ROMs

This section develops the block-structured Operator Inference method which is the main methodological contribution of this paper. Sec. II.A establishes the aeroelastic modeling problem as a representative multiphysics system. Sec. II.B reviews the standard, monolithic Operator Inference method. Finally, Sec. II.C presents the block-structured Operator Inference approach.

A. Coupled multiphysics system structure

We present the aeroelastic modeling problem as a representative example of a block-structured multiphysics system. The block-structured Operator Inference method is applicable to any multiphysics system that contains coupling in an analogous block-structured sense, but the aeroelastic setting will be used throughout the paper for clarity of notation. Aeroelastic models couple the governing equations of structural dynamics and fluid dynamics together into a single multiphysics dynamical system. The evolution of the state of this system over time is dependent both on the single-physics governing equations and on the manner in which the single-physics governing equations are coupled

together. In this paper, the structural dynamics system, fluid dynamics system, and the form of the coupling between the two systems are all permitted to be at most polynomially nonlinear. For cases with non-polynomial nonlinear terms, one could employ lifting transformations to introduce auxiliary variables that transform the system to polynomial form (see [38, 39]).

1. Structural dynamics

Structural dynamics are often modeled via linear methods, most notably an eigenvalue decomposition of the governing semi-discrete differential equations (typically derived via a finite element spatial discretization). However, to keep our initial notation general for other systems and methods, we also allow for a constant term and polynomial (here, quadratic) nonlinearities in our structural dynamics governing equations. This leads us to write the general semi-discrete system governing the structural dynamics as

$$\dot{\mathbf{q}}_s = \mathbf{c}_s + \mathbf{A}_s \mathbf{q}_s + \mathbf{H}_s (\mathbf{q}_s \otimes \mathbf{q}_s) + \mathbf{F}_s, \quad (1)$$

where $\mathbf{q}_s(t) \in \mathbb{R}^{n_s}$ is the structural state and n_s is the dimension of the structural semi-discrete first-order system of ODEs. The operators $\mathbf{c}_s \in \mathbb{R}^{n_s}$, $\mathbf{A}_s \in \mathbb{R}^{n_s \times n_s}$, and $\mathbf{H}_s \in \mathbb{R}^{n_s \times n_s^2}$ are the constant, linear, and quadratic structural dynamics operators, respectively, $\mathbf{F}_s(t) \in \mathbb{R}^{n_s}$ is the external forcing term, and \otimes denotes the Kronecker product. Note that we employ the compact version of the Kronecker product in the implementation, to avoid computing redundant terms.

2. Fluid dynamics

We similarly employ a quadratic form for the fluid dynamics semi-discrete governing equations:

$$\dot{\mathbf{q}}_f = \mathbf{c}_f + \mathbf{A}_f \mathbf{q}_f + \mathbf{H}_f (\mathbf{q}_f \otimes \mathbf{q}_f) + \mathbf{F}_f, \quad (2)$$

where $\mathbf{q}_f(t) \in \mathbb{R}^{n_f}$ is the fluid state and n_f is the dimension of the fluid dynamics semi-discrete system of ODEs. The operators $\mathbf{c}_f \in \mathbb{R}^{n_f}$, $\mathbf{A}_f \in \mathbb{R}^{n_f \times n_f}$, and $\mathbf{H}_f \in \mathbb{R}^{n_f \times n_f^2}$ are the constant, linear, and quadratic fluid dynamics operators, respectively, and $\mathbf{F}_f(t) \in \mathbb{R}^{n_f}$ is the external forcing term. As shown in [39], a variety of fluid dynamics models can be written in this quadratic form, particularly with an appropriate choice of fluid state variables. Even for the nonlinear Navier-Stokes equations, there are many situations in which a spatial discretization that yields primarily quadratic nonlinearities can be achieved (one such strategy will be employed in the practical application in Sec. III).

3. Aeroelastic dynamics

In keeping with the form established for (1) and (2), we express the coupling via the external forcing terms, $\mathbf{F}_s(t)$ and $\mathbf{F}_f(t)$. We can assume various forms for these terms, and thus model linear, bilinear, and quadratic coupling. We

could also model higher-order polynomial coupling, but for clarity of presentation here we stop at second-order coupling. Thus we write the explicit form of the coupling-specific forcing terms as

$$\mathbf{F}_s = \mathbf{E}_s \mathbf{q}_f + \mathbf{L}_s(\mathbf{q}_s \otimes \mathbf{q}_f) + \mathbf{G}_s(\mathbf{q}_f \otimes \mathbf{q}_f) \quad (3)$$

$$\mathbf{F}_f = \mathbf{E}_f \mathbf{q}_s + \mathbf{L}_f(\mathbf{q}_s \otimes \mathbf{q}_f) + \mathbf{G}_f(\mathbf{q}_s \otimes \mathbf{q}_s), \quad (4)$$

where \mathbf{E}_s and \mathbf{E}_f are the linear coupling operators, \mathbf{L}_s and \mathbf{L}_f are the bilinear coupling operators, and \mathbf{G}_s and \mathbf{G}_f are the quadratic coupling operators. Each of these operators acts on quantities that are computed using the state from the other physics component, thus providing the coupling.

Substituting the definitions in (3) and (4) into the single-physics governing equations (1) and (2), we can combine the equations into a single block-structured multiphysics system of ODEs. This yields the full-order system of multiphysics (aeroelastic) semi-discrete governing equations,

$$\begin{bmatrix} \dot{\mathbf{q}}_s \\ \dot{\mathbf{q}}_f \end{bmatrix} = \begin{bmatrix} \mathbf{c}_s \\ \mathbf{c}_f \end{bmatrix} + \begin{bmatrix} \mathbf{A}_s & \mathbf{E}_s \\ \mathbf{E}_f & \mathbf{A}_f \end{bmatrix} \begin{bmatrix} \mathbf{q}_s \\ \mathbf{q}_f \end{bmatrix} + \begin{bmatrix} \mathbf{H}_s & \mathbf{L}_s & \mathbf{G}_s \\ \mathbf{G}_f & \mathbf{L}_f & \mathbf{H}_f \end{bmatrix} \begin{bmatrix} (\mathbf{q}_s \otimes \mathbf{q}_s) \\ (\mathbf{q}_s \otimes \mathbf{q}_f) \\ (\mathbf{q}_f \otimes \mathbf{q}_f) \end{bmatrix}. \quad (5)$$

For use in later comparisons to standard Operator Inference, note that (5) can be written compactly in monolithic form as

$$\dot{\mathbf{q}} = \mathbf{c} + \mathbf{A}\mathbf{q} + \mathbf{H}(\mathbf{q} \otimes \mathbf{q}), \quad (6)$$

where $\mathbf{q} = \begin{bmatrix} \mathbf{q}_s^\top & \mathbf{q}_f^\top \end{bmatrix}^\top \in \mathbb{R}^n$ is the coupled state vector, $n = n_s + n_f$ is the full order state dimension, and $\mathbf{c} \in \mathbb{R}^n$, $\mathbf{A} \in \mathbb{R}^{n \times n}$, and $\mathbf{H} \in \mathbb{R}^{n \times n^2}$ are respectively the constant, linear, and quadratic monolithic operators.

B. Monolithic Operator Inference

Operator Inference is a nonintrusive, physics-informed reduced-order modeling method that takes advantage of the structure-preserving properties of projection-based model reduction. However, instead of intrusively projecting the semi-discrete operators to the reduced subspace, Operator Inference formulates and solves a linear least squares problem to nonintrusively infer the reduced operators from training data consisting of snapshots of the full-order system state as it is evolved in time [18]. Here we present an overview of the standard (monolithic) Operator Inference method, applicable to monolithic (single-physics) model-reduction applications (i.e., to systems governed by dynamics of the form of (6) that ignore any internal block structure).

The snapshot matrix is assembled by concatenating k_{train} full-order training snapshot vectors into a matrix $\mathbf{Q} = [\mathbf{q}_1 \ \mathbf{q}_2 \ \cdots \ \mathbf{q}_{k_{\text{train}}}] \in \mathbb{R}^{n \times k_{\text{train}}}$, where each snapshot \mathbf{q}_i with $i = 1, 2, \dots, k_{\text{train}}$ corresponds to a single column

and k_{train} refers to the number of training snapshots. To reduce the dimensionality of the problem, we use the proper orthogonal decomposition (POD) of the full-order snapshot matrix \mathbf{Q} to identify a basis which spans the desired reduced subspace. POD proceeds by computing the thin singular value decomposition as $\mathbf{Q} = \mathbf{V}\mathbf{\Sigma}\mathbf{W}^\top$, where $\mathbf{V} \in \mathbb{R}^{n \times k_{\text{train}}}$, $\mathbf{\Sigma} \in \mathbb{R}^{k_{\text{train}} \times k_{\text{train}}}$, and $\mathbf{W} \in \mathbb{R}^{k_{\text{train}} \times k_{\text{train}}}$. We then select the first r left singular vectors \mathbf{v}_j with $j = 1, 2, \dots, r$ which correspond to the first r columns of \mathbf{V} , to give the basis matrix $\mathbf{V}_r = [\mathbf{v}_1 \ \mathbf{v}_2 \ \dots \ \mathbf{v}_r] \in \mathbb{R}^{n \times r}$, where r is the desired reduced state dimension.

The full-order state is approximated in the reduced basis as $\mathbf{q} \approx \mathbf{V}_r \hat{\mathbf{q}}$, where $\hat{\mathbf{q}} \in \mathbb{R}^r$ is the reduced state vector. We now seek a ROM that governs the dynamics of the reduced state $\hat{\mathbf{q}}(t)$. Projection-based model reduction theory defines the form of the ROM to be

$$\dot{\hat{\mathbf{q}}} = \hat{\mathbf{c}} + \hat{\mathbf{A}}\hat{\mathbf{q}} + \hat{\mathbf{H}}(\hat{\mathbf{q}} \otimes \hat{\mathbf{q}}) \quad (7)$$

where $\hat{\mathbf{c}} \in \mathbb{R}^r$, $\hat{\mathbf{A}} \in \mathbb{R}^{r \times r}$, and $\hat{\mathbf{H}} \in \mathbb{R}^{r \times r^2}$ are the constant, linear, and quadratic monolithic reduced operators. Note that the ROM (7) has the same structure as the full-order model (6), due to the properties of approximation by projection (see [18, 19]).

We project the snapshot matrix to the reduced space defined by the POD basis, $\hat{\mathbf{Q}} = \mathbf{V}_r^\top \mathbf{Q} \in \mathbb{R}^{r \times k_{\text{train}}}$. We then nonintrusively infer the reduced operators by solving the linear least squares problem:

$$\hat{\mathbf{O}} = \arg \min_{\hat{\mathbf{O}}} \left\| \hat{\mathbf{D}}\hat{\mathbf{O}}^\top - \hat{\mathbf{R}}^\top \right\|_F^2 \quad (8)$$

where F denotes the Frobenius norm and we define

$$\begin{aligned} \hat{\mathbf{D}} &= \begin{bmatrix} \mathbf{1}_{k_{\text{train}}}^\top & \hat{\mathbf{Q}}^\top & (\hat{\mathbf{Q}} \otimes \hat{\mathbf{Q}})^\top \end{bmatrix} && \text{data matrix} \\ \hat{\mathbf{O}} &= \begin{bmatrix} \hat{\mathbf{c}} & \hat{\mathbf{A}} & \hat{\mathbf{H}} \end{bmatrix} && \text{operator matrix} \\ \hat{\mathbf{R}} &= \begin{bmatrix} \dot{\hat{\mathbf{Q}}} \end{bmatrix} && \text{right-hand-side matrix} \end{aligned}$$

where $\mathbf{1}_{k_{\text{train}}} \in \mathbb{R}^{1 \times k_{\text{train}}}$ is a k_{train} -dimensional unity row vector. If the time derivatives from the solver are available, then we populate the right-hand-side matrix, $\hat{\mathbf{R}}$, directly from the reduced time derivative snapshot matrix provided by the flow solver, $\dot{\hat{\mathbf{Q}}} = \mathbf{V}_r^\top \dot{\mathbf{Q}}$. However, these values are often not available when working with commercial or restricted solvers (such as FUN3D, which we use for the numerical example beginning in Sec. III). Thus, our recourse in this work is to compute $\dot{\hat{\mathbf{Q}}}$ via a sixth-order centered finite difference approximation. Note that we compute the finite differences in the reduced subspace, not the full space, to avoid the additional computational cost of a second projection operation.

For many real-world Operator Inference applications with high-dimensional full-order models and complex dynamics, the stability of the inferred ROMs can be a challenge. This is due to the use of sparse and noisy training data, which,

when compounded with ill-conditioned least squares problems, can lead to overfitting of the operators. Therefore, adding a regularizer \mathcal{R} to (8) is typically needed to improve the robustness of the learning step at the expense of the possible loss of some accuracy [40]. Using a Tikhonov regularization function $\mathcal{R}(\widehat{\mathbf{O}}) = \gamma_c \|\widehat{\mathbf{C}}\|_F^2 + \gamma_A \|\widehat{\mathbf{A}}\|_F^2 + \gamma_H \|\widehat{\mathbf{H}}\|_F^2$, the least squares problem then becomes

$$\widehat{\mathbf{O}} = \arg \min_{\widehat{\mathbf{O}}} \left(\|\widehat{\mathbf{D}}\widehat{\mathbf{O}}^\top - \widehat{\mathbf{R}}^\top\|_F^2 + \gamma_c \|\widehat{\mathbf{C}}\|_F^2 + \gamma_A \|\widehat{\mathbf{A}}\|_F^2 + \gamma_H \|\widehat{\mathbf{H}}\|_F^2 \right), \quad (9)$$

where γ_c , γ_A , and γ_H are scalar regularization hyperparameters that we choose to separately weight the penalization of each operator. In practice, for Operator Inference these regularization hyperparameters are identified by performing a grid search to identify the combination of regularization levels that leads to ROM operators which most accurately predict the true training snapshot trajectory [40]. One may also incorporate a bounded growth constraint [40] to further encourage the stability of the resulting ROMs as described in Appendix A of [41]. This constraint requires that the deviations of each reduced state do not exceed the training regime deviations by more than a user-determined factor, α , which becomes another hyperparameter for the learning problem that can be identified via a similar grid search.

C. Block-structured Operator Inference

Block-structured Operator Inference uses multiple instances of the monolithic Operator Inference approach presented in Sec. II.B to decompose a block-structured model reduction problem into multiple smaller subproblems. These subproblems are related to each other via the coupling mechanisms discussed in Sec. II.A. The single-physics operators corresponding to each physics regime (e.g., structural dynamics and fluid dynamics) are inferred via separate least squares subproblems. This separation allows for added flexibility in regularizing the operators, nonintrusively imposing structure, and intrusively imposing any known operators for each physics regime.

To define the decomposition of the system into subproblems, consider the multiphysics governing equations with block structure given by (5). Separate bases $\mathbf{V}_s \in \mathbb{R}^{r_s \times n_s}$ and $\mathbf{V}_f \in \mathbb{R}^{r_f \times n_f}$ must be chosen for each subproblem (e.g., via POD), where r_s and r_f are respectively the dimensions of the reduced systems of ODEs for the structural dynamics and fluid dynamics. These bases are combined in block-diagonal form into a coupled basis

$$\mathbf{V}_r = \begin{bmatrix} \mathbf{V}_s \\ \mathbf{V}_f \end{bmatrix} \quad (10)$$

in order to maintain the desired structure-preserving behavior [42]. This form of the coupled basis leads to an analogous structure-preserving projection property to that discussed in Sec. II.B, which in turn means that the resulting multiphysics

ROM structure is

$$\begin{bmatrix} \dot{\hat{\mathbf{q}}}_s \\ \dot{\hat{\mathbf{q}}}_f \end{bmatrix} = \begin{bmatrix} \hat{\mathbf{c}}_s \\ \hat{\mathbf{c}}_f \end{bmatrix} + \begin{bmatrix} \hat{\mathbf{A}}_s & \hat{\mathbf{E}}_s \\ \hat{\mathbf{E}}_f & \hat{\mathbf{A}}_f \end{bmatrix} \begin{bmatrix} \hat{\mathbf{q}}_s \\ \hat{\mathbf{q}}_f \end{bmatrix} + \begin{bmatrix} \hat{\mathbf{H}}_s & \hat{\mathbf{L}}_s & \hat{\mathbf{G}}_s \\ \hat{\mathbf{G}}_f & \hat{\mathbf{L}}_f & \hat{\mathbf{H}}_f \end{bmatrix} \begin{bmatrix} (\hat{\mathbf{q}}_s \otimes \hat{\mathbf{q}}_s) \\ (\hat{\mathbf{q}}_s \otimes \hat{\mathbf{q}}_f) \\ (\hat{\mathbf{q}}_f \otimes \hat{\mathbf{q}}_f) \end{bmatrix}, \quad (11)$$

which matches the structure of the full-order block-structured governing ODEs in (5). In other words, our block-structured Operator Inference approach yields a ROM that retains the block structure of the original multiphysics full-order system.

To determine the ROM operators, we observe that the operators for each physics component can be inferred via a separate least squares process of an analogous form to (9). This is due to the separability of each column of a matrix least squares problem into an independent vector least squares problem (see [18] for details). Therefore, for the structural dynamics subproblem, we solve

$$\hat{\mathbf{O}}_s = \arg \min_{\hat{\mathbf{O}}_s} \left\| \hat{\mathbf{D}}_s \hat{\mathbf{O}}_s^\top - \hat{\mathbf{R}}_s^\top \right\|_F^2 + \mathcal{R}(\hat{\mathbf{O}}_s), \quad (12)$$

where we define

$$\begin{aligned} \hat{\mathbf{D}}_s &= \begin{bmatrix} \mathbf{1}_k^\top & \hat{\mathbf{Q}}_s^\top & \hat{\mathbf{Q}}_f^\top & (\hat{\mathbf{Q}}_s \otimes \hat{\mathbf{Q}}_s)^\top & (\hat{\mathbf{Q}}_s \otimes \hat{\mathbf{Q}}_f)^\top & (\hat{\mathbf{Q}}_f \otimes \hat{\mathbf{Q}}_f)^\top \end{bmatrix} && \text{data matrix} \\ \hat{\mathbf{O}}_s &= \begin{bmatrix} \hat{\mathbf{c}}_s & \hat{\mathbf{A}}_s & \hat{\mathbf{E}}_s & \hat{\mathbf{H}}_s & \hat{\mathbf{L}}_s & \hat{\mathbf{G}}_s \end{bmatrix} && \text{operator matrix} \\ \hat{\mathbf{R}}_s &= \begin{bmatrix} \dot{\hat{\mathbf{Q}}}_s \end{bmatrix} && \text{right-hand-side matrix} \end{aligned}$$

For the fluid dynamics subproblem, we have

$$\hat{\mathbf{O}}_f = \arg \min_{\hat{\mathbf{O}}_f} \left\| \hat{\mathbf{D}}_f \hat{\mathbf{O}}_f^\top - \hat{\mathbf{R}}_f^\top \right\|_F^2 + \mathcal{R}(\hat{\mathbf{O}}_f), \quad (13)$$

where we define

$$\begin{aligned} \hat{\mathbf{D}}_f &= \begin{bmatrix} \mathbf{1}_k^\top & \hat{\mathbf{Q}}_s^\top & \hat{\mathbf{Q}}_f^\top & (\hat{\mathbf{Q}}_s \otimes \hat{\mathbf{Q}}_s)^\top & (\hat{\mathbf{Q}}_s \otimes \hat{\mathbf{Q}}_f)^\top & (\hat{\mathbf{Q}}_f \otimes \hat{\mathbf{Q}}_f)^\top \end{bmatrix} && \text{data matrix} \\ \hat{\mathbf{O}}_f &= \begin{bmatrix} \hat{\mathbf{c}}_f & \hat{\mathbf{E}}_f & \hat{\mathbf{A}}_f & \hat{\mathbf{G}}_f & \hat{\mathbf{L}}_f & \hat{\mathbf{H}}_f \end{bmatrix} && \text{operator matrix} \\ \hat{\mathbf{R}}_f &= \begin{bmatrix} \dot{\hat{\mathbf{Q}}}_f \end{bmatrix} && \text{right-hand-side matrix} \end{aligned}$$

Note that in this general formulation, $\hat{\mathbf{D}}_s = \hat{\mathbf{D}}_f$, and the structural and fluid Operator Inference problems differ only in the right-hand side matrices since $\hat{\mathbf{R}}_s \neq \hat{\mathbf{R}}_f$. However, for specific problems, the multiphysics ROM (11) may not have all blocks populated, which would lead to $\hat{\mathbf{D}}_s \neq \hat{\mathbf{D}}_f$. As we will show in Sec. III, introducing problem-specific structure into the form of the ROM is part of the utility and desirability of the block-structured approach.

Regularization for each least squares subproblem can be achieved in the same manner as for the monolithic case in Sec. II.B. The scales of the regularization values may differ for each subproblem. This gives an opportunity to better tune the regularization strategy, but also provides a potential computational challenge, since the increasing dimensionality of the regularization parameters leads to increased computational complexity in determining optimal regularization levels.

III. Aeroelastic model of the AGARD 445.6 wing

The AGARD 445.6 wing [22] is a commonly used aeroelastic modeling validation case. Experimental flutter testing was conducted on the AGARD wing at NASA’s Transonic Dynamics Tunnel [22] and the results of those tests have become a canonical flutter modeling validation dataset. This section describes the full-order numerical model that will generate train and test data sets to be used to study the Operator Inference ROMs. Section III.A describes the geometry and spatial discretizations of the AGARD 445.6 wing, and Sec. III.B presents the full-order aeroelastic modeling approach as implemented in FUN3D to couple the fluid dynamics and structural dynamics.

A. Geometry and discretization

The AGARD wing is a half-span swept and tapered wing with a root chord of 22 inches, a tip chord of 14.5 inches, a taper ratio of 0.6576, an aspect ratio of 1.6525, a span of 30 inches, and a quarter-chord sweepback angle of 45 deg. It has a symmetric NACA65A004 airfoil and is made of laminated mahogany that was deliberately weakened by drilling holes and filling them with foam, inducing heightened flexibility during wind tunnel testing [22].

The CFD grid used in this paper^{*} was designed for a viscous, turbulent finite volume CFD model and has $g_f = 3,613,171$ grid points. The structural dynamics grid[†] uses two-dimensional plate finite elements and has $g_s = 121$ grid points.

B. Numerical model

We use NASA’s FUN3D solver as the full-order model. FUN3D is a fully-unstructured Navier-Stokes finite volume solver for CFD [37]. It has an integrated aeroelasticity capability that couples the high-fidelity nonlinear fluid dynamics solution to a linear structural dynamics solver based on an eigenvalue modal decomposition [43]. Note that we refer to this setup as our full-order model, even though the structural dynamics model is already a reduced-order modal model. This eigenvalue modal decomposition approach, which will be described in detail in Sec. III.B.2, uses the eigenvectors of the finite element model to span the structural reduced basis \mathbf{V}_s from (10) and thus provide us with a structural dynamics ROM. This is acceptable in the context of aeroelastic modeling because the fluid dynamics solution typically requires a significant majority of the computational resources and is therefore the component of the solution that is of most interest in a model reduction context. Here we first present the fluid dynamics model (Sec. III.B.1), then the

^{*}Provided by Pawel Chwalowski of the NASA Langley Research Center.

[†]FUN3D v13.4 Training - Session 16: Aeroelastic Simulations: https://fun3d.larc.nasa.gov/session16_2018.pdf

structural dynamics model (Sec. III.B.2), and finally the coupling between the two models (Sec. III.B.3).

1. Fluid dynamics

We use FUN3D to solve the unsteady, compressible, turbulent Navier-Stokes equations for flow over the AGARD 445.6 wing, starting from the form

$$\frac{\partial \rho}{\partial t} + \nabla \cdot \rho \vec{u} = 0 \quad (14a)$$

$$\frac{\partial \rho \vec{u}}{\partial t} + \nabla \cdot (\rho \vec{u} \otimes \vec{u} - \sigma) = 0 \quad (14b)$$

$$\frac{\partial e}{\partial t} + \nabla \cdot (e \vec{u} - k_{\text{thermal}} \nabla T - \sigma \cdot \vec{u}) = 0 \quad (14c)$$

where $\sigma = -p\mathbf{I} + \mu \left[-\frac{2}{3}(\nabla \cdot \vec{u})\mathbf{I} + \nabla \vec{u} + (\nabla \vec{u})^\top \right]$ is the diffusive flux tensor, p is the pressure, ρ is the density, $\vec{u} = [u \ v \ w]$ are the fluid velocity components in each coordinate direction, e is the energy per unit volume, and k_{thermal} is the thermal conductivity. One must also specify suitable initial conditions and boundary conditions.

We assume a thermally and calorically perfect gas with equation of state

$$T = \frac{p}{\rho R} \quad (15)$$

where T is the temperature and $R = 1716.49 \text{ ft-lbf/slug-}^\circ\text{R}$ is the specific gas constant for air [44]. The speed of sound is defined by the relation

$$a^2 = \gamma_{\text{specific}} RT \quad (16)$$

where $\gamma_{\text{specific}} = 1.4$ is the specific heat ratio for air. The specific heat relations are

$$\gamma_{\text{specific}} = \frac{c_p}{c_v} \quad (17)$$

$$c_p - c_v = R \quad (18)$$

$$\frac{R}{c_p} = \frac{\gamma_{\text{specific}} - 1}{\gamma_{\text{specific}}} \quad (19)$$

where c_p and c_v are the specific heat capacities at constant pressure and constant volume. Finally, we define the relationship between temperature and viscosity via Sutherland's law

$$\mu = \mu_{\text{ref}} \frac{T_{\text{ref}} + C}{T + C} \left(\frac{T}{T_{\text{ref}}} \right)^{3/2} \quad (20)$$

where $C = 198.6 \text{ }^\circ\text{R}$ for air [37], $T_{\text{ref}} = 518.69 \text{ }^\circ\text{R}$ (assuming standard atmosphere at sea level), and $\mu_{\text{ref}} = 3.737E - 7 \text{ slug/ft-s}$ (assuming standard atmosphere at sea level) [45].

To close the RANS equations for turbulent (viscous) flow, we use the one equation Spalart-Allmaras turbulence model [46]. Adding this turbulence model means that $\mu = \mu_l + \mu_t$, where μ_l is the laminar viscosity computed via Sutherland's law and μ_t is the turbulent eddy viscosity computed via the solution of the Spalart-Allmaras turbulence equation. In practice, FUN3D solves the nondimensional form of (14). Thus, instead of assuming one of the dimensional parameters listed above, we close the nondimensional governing equations by assuming a Prandtl number of $P_r = 0.72$ for air, where $P_r = \frac{c_p \mu}{k_{\text{thermal}}}$. See [37] for further details about the nondimensionalization.

FUN3D uses the finite volume method to spatially discretize (14) and outputs the discrete solution in terms of primitive variables: pressure, \mathbf{p} , component fluid velocities, \mathbf{u} , \mathbf{v} , and \mathbf{w} , and density ρ . We use the g_f -dimensional grid mentioned above, so with five fluid states at each grid point, the full-order CFD model has $n_f = 5g_f = 18,065,855$ fluid degrees of freedom for the Navier-Stokes equations plus another $g_f = 3,613,171$ degrees of freedom for the turbulence model. For the purposes of Operator Inference, we target the form of the PDE governing equations rather than their discretized RANS form, thus we will neglect the turbulence variable's degrees of freedom in our ROM formulation. See Sec. IV.B for more details. The effects of the structural motion of the wing enter through boundary conditions, which in turn introduce the external forcing term $\mathbf{F}_f(t)$ on the right-hand-side of the semi-discrete fluid dynamics governing equation (2).

2. Structural dynamics

The finite element method is used to spatially discretize the AGARD wing geometry and generate a second order, linear system of governing equations for the structural dynamics of the form

$$\mathbf{M}\ddot{\boldsymbol{\delta}} + \mathbf{C}\dot{\boldsymbol{\delta}} + \mathbf{K}\boldsymbol{\delta} = \mathbf{F}_\delta \quad (21)$$

where $\boldsymbol{\delta}$ is the state vector of displacements, \mathbf{M} , \mathbf{C} , and \mathbf{K} are the mass, damping, and stiffness matrices, respectively, and \mathbf{F}_δ is the external input forcing term. To identify a reduced basis for the structural dynamics, we compute an eigenvalue modal decomposition via the generalized eigenvalue problem

$$\mathbf{K}\boldsymbol{\phi}_i = \omega_i^2 \mathbf{M}\boldsymbol{\phi}_i \quad (22)$$

where $\boldsymbol{\phi}_i$ is the eigenvector corresponding to the i^{th} eigenvalue ω_i . Each eigenvector $\boldsymbol{\phi}_i$ corresponds to a global structural mode shape (reduced degree of freedom), of which the first four are shown in Fig. 1 and are taken from [22] (formatted for FUN3D in the FUN3D tutorial materials[‡]).

Concatenating the eigenvectors (i.e., the basis vectors) produces a basis matrix $\boldsymbol{\Phi} \in \mathbb{R}^{n_s \times n_s}$, which can then be truncated to the desired reduced-order dimension r_s by keeping only the first r_s columns. We denote this reduced

[‡]FUN3D v13.4 Training - Session 16: Aeroelastic Simulations: https://fun3d.larc.nasa.gov/session16_2018.pdf

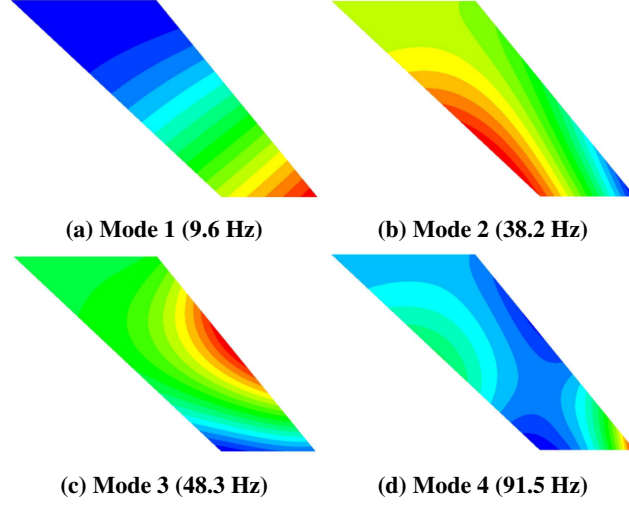


Fig. 1 Structural mode shapes of the AGARD wing, scaled independently for visualization.

basis matrix $\mathbf{V}_s \in \mathbb{R}^{n_s \times r_s}$ to indicate that it spans the reduced subspace for the structural dynamics. Now the structural dynamics equation is reduced via Galerkin projection by defining the reduced structural state $\boldsymbol{\eta}$ via the projection $\boldsymbol{\delta} = \mathbf{V}_s \boldsymbol{\eta}$ and left multiplying by \mathbf{V}_s^\top to get

$$\mathbf{V}_s^\top \mathbf{M} \mathbf{V}_s \ddot{\boldsymbol{\eta}} + \mathbf{V}_s^\top \mathbf{C} \mathbf{V}_s \dot{\boldsymbol{\eta}} + \mathbf{V}_s^\top \mathbf{K} \mathbf{V}_s \boldsymbol{\eta} = \mathbf{V}_s^\top \mathbf{F}_\delta. \quad (23)$$

Due to the orthogonality properties of the eigenvectors, we have $\mathbf{V}_s^\top \mathbf{M} \mathbf{V}_s = \mathbf{I}$, $\mathbf{V}_s^\top \mathbf{C} \mathbf{V}_s = \boldsymbol{\zeta}$, $\mathbf{V}_s^\top \mathbf{K} \mathbf{V}_s = \boldsymbol{\Omega}$, and $\mathbf{F}_\eta = \mathbf{V}_s^\top \mathbf{F}_\delta$, so we simplify to obtain

$$\ddot{\boldsymbol{\eta}} + \boldsymbol{\zeta} \dot{\boldsymbol{\eta}} + \boldsymbol{\Omega} \boldsymbol{\eta} = \mathbf{F}_\eta \quad (24)$$

where $\boldsymbol{\zeta} = \text{diag}(2\omega_1\zeta_1, 2\omega_2\zeta_2, \dots, 2\omega_{n_s}\zeta_{n_s})$ and $\boldsymbol{\Omega} = \text{diag}(\omega_1^2, \omega_2^2, \dots, \omega_{n_s}^2)$ are diagonal matrices. Then we convert the system of uncoupled scalar second-order equations to a system of vector first order equations by redefining the reduced structural state as $\widehat{\mathbf{q}}_s = [\boldsymbol{\eta}^\top \ \dot{\boldsymbol{\eta}}^\top]^\top$. This produces a first-order, block-structured system of ODEs as desired, written as

$$\begin{bmatrix} \dot{\boldsymbol{\eta}} \\ \ddot{\boldsymbol{\eta}} \end{bmatrix} = \begin{bmatrix} \mathbf{0} & \mathbf{I} \\ -\boldsymbol{\Omega} & -\boldsymbol{\zeta} \end{bmatrix} \begin{bmatrix} \boldsymbol{\eta} \\ \dot{\boldsymbol{\eta}} \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ \mathbf{F}_\eta \end{bmatrix} \quad (25)$$

where for this work, we assume zero structural damping for all of the modes, $\zeta_i = 0.0$. This system of ODEs can be written compactly as

$$\dot{\widehat{\mathbf{q}}}_s = \widehat{\mathbf{A}}_s \widehat{\mathbf{q}}_s + \widehat{\mathbf{F}}_s. \quad (26)$$

3. Aeroelastic coupling

FUN3D’s aeroelasticity capability couples the modal structural dynamics model and the RANS CFD model to create an aeroelastic multiphysics simulation [43]. The inputs from the fluid dynamics to the structural dynamics enter via the generalized aerodynamic forces \mathbf{F}_s from a given timestep of the CFD solution. Conversely, the inputs from the structural dynamics to the fluid dynamics enter via specification of fluid velocities at the fluid-solid interfaces, along with a linear elasticity mesh deformation approach which modifies the spatial discretization before each new timestep of the CFD solution. We refer the reader to [37] and [43] for further details regarding the FUN3D and aeroelasticity implementations.

IV. Block-structured Operator Inference for the AGARD 445.6 wing

Section IV.A summarizes the flow conditions used to generate full-order model training data for the Operator Inference ROM. Section IV.B describes the preprocessing steps performed to prepare the fluid training data for Operator Inference, in particular the state variable transformation, shifting, and scaling of the snapshots. Section IV.C studies the singular value decomposition of the fluid dynamics snapshots and selects a reduced dimension r_f for the fluid portion of the ROM. Section IV.D discusses the particular specializations for the AGARD wing that are enabled by the use of block-structured Operator Inference and describes the regularization grid searches performed to identify the optimal regularization levels. Finally, Sec. IV.E analyzes the performance of the block-structured Operator Inference ROM and is the main result of the paper.

A. Data generation

To generate training data from the AGARD wing aeroelastic model at varying flow conditions, we select Mach numbers in the set $\mathcal{M} = \{0.901, 0.957, 1.141\}$ and dynamic pressures in the set $\mathcal{Q} = \{50, 70, 90\}$ psf, for a total of nine flow conditions and thus nine training data sets. The Mach numbers match the canonical experimental test conditions from [22] and the dynamic pressures provide a uniform grid of values analogous to what might be selected naively for flutter boundary characterization. We initially included higher dynamic pressures (above 90 psf), but we found that for some flow conditions FUN3D failed to solve at those higher levels. Thus we have limited ourselves to a range of dynamic pressures for which we were able to solve for all flow conditions successfully, allowing us to maintain a uniform grid of training set flow conditions. The training set flow conditions are plotted in Fig. 2 along with some experimental and computational flutter points to illustrate the expected behavior at various flow conditions.

The FUN3D full-order model is simulated at each flow condition for 1000 total timesteps, corresponding to a final time of $t_f = 0.2453$ seconds, thus providing the full data set for use across training and testing. Preliminary testing suggested that this would be a sufficient number of timesteps to provide useful training data and reserve a significant time period from each response for testing of the learned ROMs. For each flow condition (Mach number and

dynamic pressure pair), all four structural generalized velocities are perturbed simultaneously by $gvel_i = 0.1$ while the structural generalized displacements are kept at $gdisp_i = 0$, and then the dynamics are allowed to evolve from that initial perturbation to assess the aeroelastic response. See Appendix for a detailed description of how the FUN3D input parameters were computed for each flow condition.

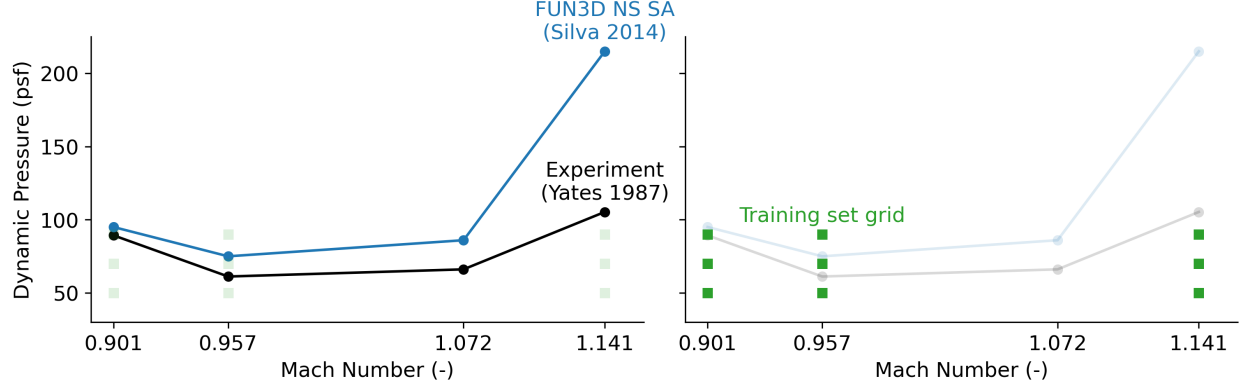


Fig. 2 Flutter boundaries from experiments [22] and modeling [47] in the literature (left) and training set flow conditions (right) for the AGARD 445.6 wing.

B. Fluid data preprocessing

Operator Inference learns ROMs for dynamics with governing equations that have polynomial form and neglects any remaining non-polynomial nonlinear behavior. Although the Navier-Stokes equations do not satisfy this requirement in their primitive state variable form, previous work has shown that transforming the fluid state to specific volume form induces a quadratic form [39, 48]. We note that this quadratic form applies to the PDE governing equations and not necessarily their discretized form. In particular, the Spalart-Allmaras turbulence model has fourth-order dynamics, and so an intrusively-derived projection-based ROM that perfectly mimicked the FUN3D discretization would include higher order terms. Nonetheless, transforming the snapshots to specific volume variables means that we can target an Operator Inference ROM with quadratic form. We expect that the error incurred due to model misspecification (i.e., ignoring higher order terms) will be less than the approximations introduced by the dimension reduction itself.

FUN3D solves nondimensional governing equations derived from (14) and provides discretized state outputs in primitive form $[\mathbf{p}^\top \mathbf{u}^\top \mathbf{v}^\top \mathbf{w}^\top \boldsymbol{\rho}^\top]^\top$, so converting to specific volume variables can be accomplished via a simple postprocessing of the snapshots. This postprocessing replaces the density $\boldsymbol{\rho}$ with the specific volume $\boldsymbol{\zeta}$ via the transformation $\boldsymbol{\zeta} = 1/\boldsymbol{\rho}$. We emphasize that this variable transformation is applied only to the snapshot data, and not to FUN3D itself. Thus, as shown in [41], we employ a state vector for Operator Inference of the form $\mathbf{q}_f = [\mathbf{p}^\top \mathbf{u}^\top \mathbf{v}^\top \mathbf{w}^\top \boldsymbol{\zeta}^\top]^\top$, and we learn a fluid dynamic ROM with quadratic form.

After transforming the fluid state to induce quadratic form, we preprocess the fluid snapshot data via shifting and scaling to encourage more robust basis identification during POD and inference in the least squares subproblems. We

shift the states by subtracting the mean (over time) pressure, velocity, or specific volume value from each pressure, velocity, or specific volume snapshot, respectively. Then we scale the entire fluid snapshot by adjusting each fluid state variable to have pressure and velocity entries in the range of $[-1, 1]$, and specific volume entries in the range of $[0, 1]$. The full-order model data is now ready for use in the POD and inference steps of the Operator Inference method. Note that shifting the fluid snapshots changes the structure of the ROM ODE. The shifting introduces constant operator terms in addition to the original ROM operators, so the final ROM form consists of constant, linear, and quadratic operators and thus matches the form of (11).

C. Singular value decomposition of fluid snapshots

Now we use the preprocessed training data to identify an appropriate reduced basis for each ROM via the proper orthogonal decomposition (POD). We compute the SVD of the training snapshot matrix for each flow condition, where the fluid training snapshot matrix $\mathbf{Q}_f \in \mathbb{R}^{n_f \times k_{\text{train}}}$ has k_{train} columns, one for each timestep of training data. Here we study $k_{\text{train}} \in \{100, 300, 500\}$ to investigate the effects of varying the amount of training data required. This is especially important for such a high-dimensional problem because generating each additional training snapshot requires significant computational expense, so we desire that k_{train} be as small as possible.

Figure 3 shows the singular value decay for each flow condition and each studied k_{train} value. We can see that for each flow condition, the singular value decay is slower for higher k_{train} , indicating that there is more information contained in the snapshots. The singular values decay at similar rates for the first few indices across all k_{train} values, but then the $k_{\text{train}} = 100$ decay rate increases after about five singular values ($r_f = 5$) and the $k_{\text{train}} = 300$ decay rate increases after about 25 singular values ($r_f = 25$). While the figure shows that the snapshot set with $k_{\text{train}} = 500$ contains more information than the set with $k_{\text{train}} = 300$, the difference is relatively small. We thus choose to use $k_{\text{train}} = 300$ as the size of our training data set, which provides a good balance between richness of information and magnitude of training data generation costs and storage.

In contrast to the differences in the decay rates between k_{train} levels, we see negligible differences between flow conditions (changing Mach number and dynamic pressure). This suggests that similarly sized ROM bases should perform comparably well across the studied space of flow conditions, i.e., we can choose a single r_f size and use it for all nine ROMs (one ROM per flow condition).

We also plot the cumulative energy captured for each flow condition for $k_{\text{train}} = 300$ in Fig. 4 as a function of ROM basis size r_f . Each subplot calls out the reduced dimension required to capture 99.99% and 99.9995% of the cumulative energy. We see that to capture 99.99% of the cumulative energy for all nine flow conditions, we require $r_f = 8$, and to capture 99.9995% of the cumulative energy for all nine flow conditions, we require $r_f = 12$; therefore we select these two r_f levels to generate ROMs for the following ROM performance analyses. These selections will lead to a total coupled ROM dimension of $r = r_s + r_f = 8 + 8 = 16$ or $r = 8 + 12 = 20$.

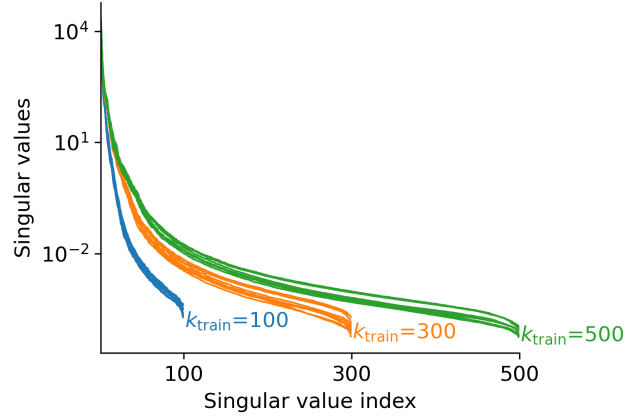


Fig. 3 Singular values for snapshot sets of size $k_{\text{train}} = 100, 300$ and 500 . For each size, a different snapshot set is analyzed for each of the nine (M_∞, q_∞) flow conditions.

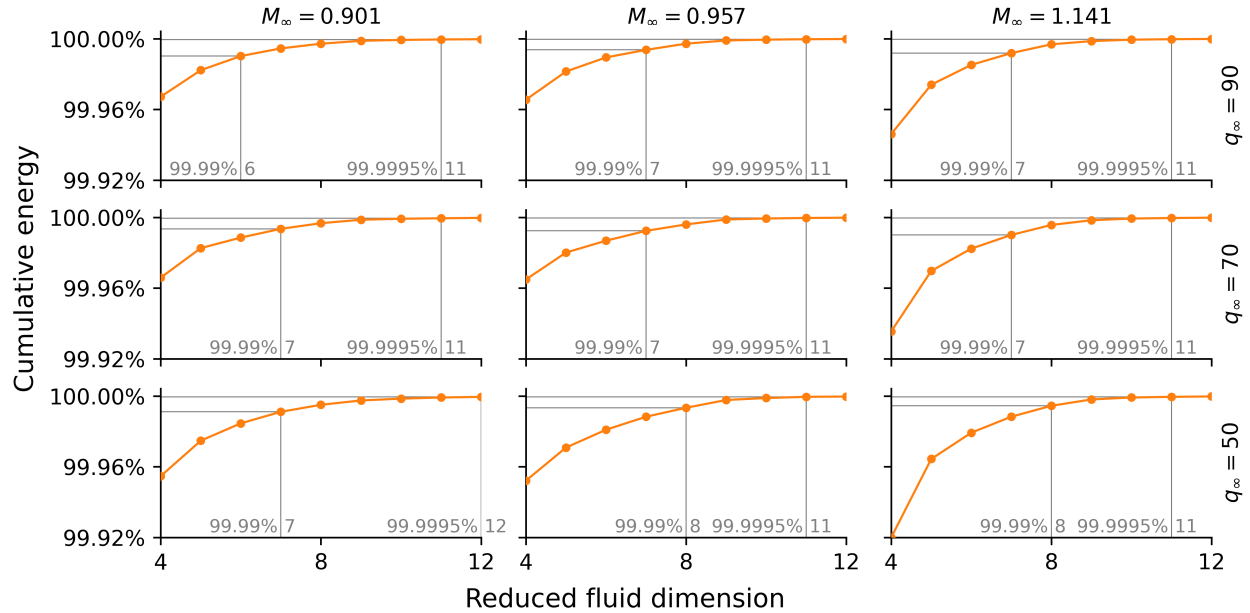


Fig. 4 Cumulative energy captured for varying M_∞ and q_∞ where $k_{\text{train}} = 300$ and thresholds are marked for at least 99.99% and 99.9995% cumulative energy captured.

We use a single reduced basis across all five fluid state variables, so each POD mode contains information about the pressure, velocities, and specific volume over the entire spatial domain of the fluid dynamics. In Fig. 5 we plot the portion of each POD mode that corresponds to the pressure on the surface of the wing for the $M_\infty = 0.901$ and $q_\infty = 50$ psf case with $k_{\text{train}} = 300$ training snapshots. Mode 1 has negligible surface pressure behavior. This is due to the chosen shifting and scaling of snapshots before computing the POD modes as described in Sec. IV.B. (Note that, in this example, with the chosen scaling the first POD mode predominantly represents variation in specific volume.) Modes 2 and 3 describe localized behavior of the surface pressure near the leading edge and tip of the wing, whereas Modes 4–12 characterize more global behavior across the wing surface.

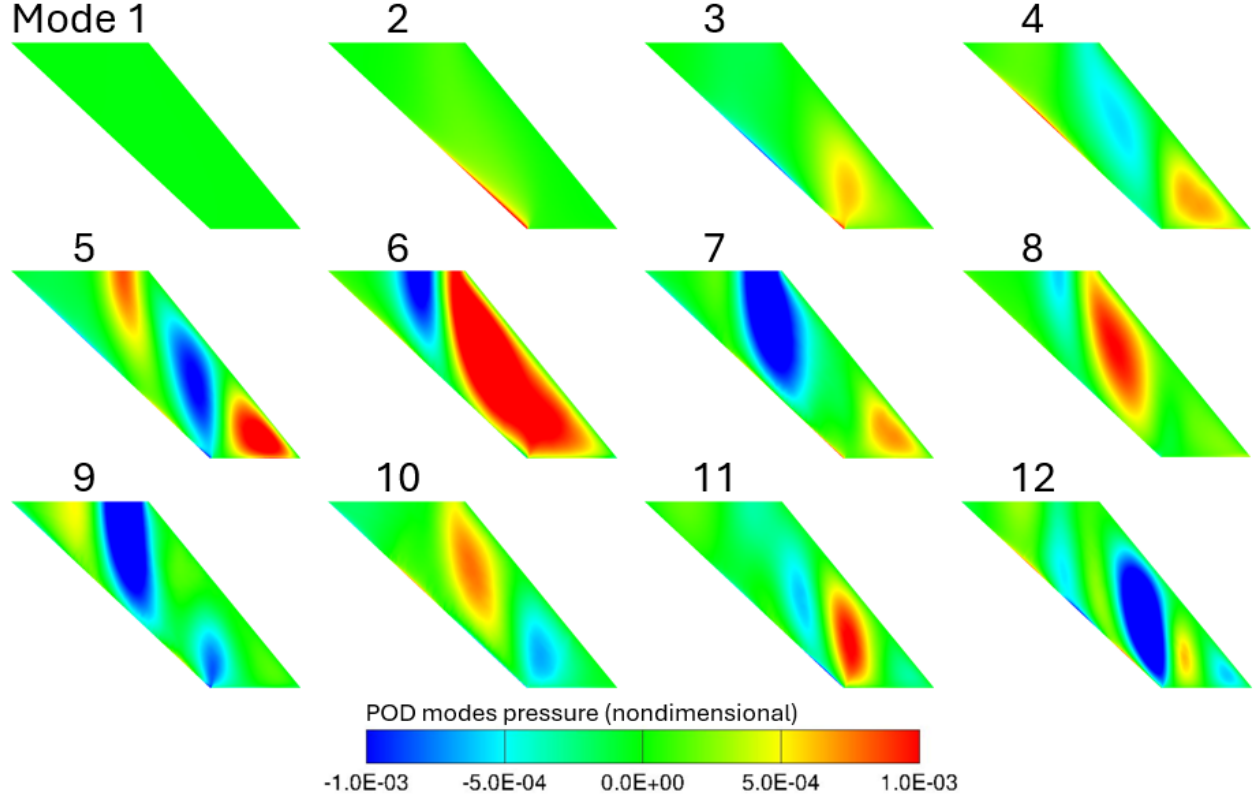


Fig. 5 Centered and scaled nondimensional surface pressure portion of each POD mode for $M_\infty = 0.901$, $q_\infty = 50$ psf, and $k_{\text{train}} = 300$.

D. Specializations enabled by the block structure

Incorporating block structure into Operator Inference provides added flexibility which we exploit to specialize the learning formulation. We use the fluid dynamics and structural dynamics basis matrices to construct a block-structured reduced basis matrix of the form in (10). This form has the block-structure-preserving property which permits us to split the monolithic (coupled) least squares problem in (9) into the single-physics least squares problems in (12) and (13) and take advantage of the block-structured specializations.

Most importantly, we recognize that the structural dynamics governing equation form in (26) is linear while, after lifting to specific volume form and considering the Navier-Stokes governing equations in their PDE form, the fluid dynamics governing equation form is quadratic. The imposition of the linear structural dynamics form can alternatively be seen as intrusively specifying the quadratic structural dynamics operator to be zero, $\hat{\mathbf{H}}_s = \mathbf{0}$. We also represent the linear coupling behavior in both the structure-to-fluid and fluid-to-structure directions. Referring back to the derivation of the block structure from Sec. III.B.3, this means that we can take advantage of the block structure of the coupling terms to impose that the bilinear and quadratic coupling operators should be zero as well, i.e., $\hat{\mathbf{L}}_s = \hat{\mathbf{G}}_s = \hat{\mathbf{G}}_f = \hat{\mathbf{L}}_f = \mathbf{0}$.

These specializations lead us to update the desired ROM form that we originally specified in (11) by removing the five zeroed-out blocks of the quadratic operator and replacing them with zero blocks. We can now write our target ROM

dynamics in the form

$$\begin{bmatrix} \dot{\hat{\mathbf{q}}}_s \\ \dot{\hat{\mathbf{q}}}_f \end{bmatrix} = \begin{bmatrix} \hat{\mathbf{c}}_s \\ \hat{\mathbf{c}}_f \end{bmatrix} + \begin{bmatrix} \hat{\mathbf{A}}_s & \hat{\mathbf{E}}_s \\ \hat{\mathbf{E}}_f & \hat{\mathbf{A}}_f \end{bmatrix} \begin{bmatrix} \hat{\mathbf{q}}_s \\ \hat{\mathbf{q}}_f \end{bmatrix} + \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \hat{\mathbf{H}}_f \end{bmatrix} \begin{bmatrix} (\hat{\mathbf{q}}_s \otimes \hat{\mathbf{q}}_s) \\ (\hat{\mathbf{q}}_s \otimes \hat{\mathbf{q}}_f) \\ (\hat{\mathbf{q}}_f \otimes \hat{\mathbf{q}}_f) \end{bmatrix}. \quad (27)$$

Crucially, this new form leads to many simplifications in the least squares subproblems from (12) and (13). We simplify the least squares input and output matrices $\hat{\mathbf{D}}_s$, $\hat{\mathbf{R}}_s$, $\hat{\mathbf{O}}_s$, $\hat{\mathbf{D}}_f$, $\hat{\mathbf{R}}_f$, and $\hat{\mathbf{O}}_f$ from Sec. II.C by removing the intrusively specified blocks and shifting any known terms to the right-hand-side matrices. However, the intrusively specified blocks in this problem are all zeros, so we just remove them from the computations. Then the structural dynamics least squares problem matrices are

$$\hat{\mathbf{D}}_s = \begin{bmatrix} \mathbf{1}_k^\top & \hat{\mathbf{Q}}_s^\top & \hat{\mathbf{Q}}_f^\top \end{bmatrix} \quad (28)$$

$$\hat{\mathbf{O}}_s = \begin{bmatrix} \hat{\mathbf{c}}_s & \hat{\mathbf{A}}_s & \hat{\mathbf{E}}_s \end{bmatrix} \quad (29)$$

$$\hat{\mathbf{R}}_s = \begin{bmatrix} \dot{\hat{\mathbf{Q}}}_s \end{bmatrix} \quad (30)$$

and the fluid dynamics least squares problem matrices are

$$\hat{\mathbf{D}}_f = \begin{bmatrix} \mathbf{1}_k^\top & \hat{\mathbf{Q}}_s^\top & \hat{\mathbf{Q}}_f^\top & (\hat{\mathbf{Q}}_f \otimes \hat{\mathbf{Q}}_f)^\top \end{bmatrix} \quad (31)$$

$$\hat{\mathbf{O}}_f = \begin{bmatrix} \hat{\mathbf{c}}_f & \hat{\mathbf{E}}_f & \hat{\mathbf{A}}_f & \hat{\mathbf{H}}_f \end{bmatrix} \quad (32)$$

$$\hat{\mathbf{R}}_f = \begin{bmatrix} \dot{\hat{\mathbf{Q}}}_f \end{bmatrix}. \quad (33)$$

Finally, the separation of the learning step into two least squares subproblems permits us to separately regularize each block in the operator matrices $\hat{\mathbf{O}}_s$ and $\hat{\mathbf{O}}_f$. In principle, we could choose independent regularization parameters for each operator $\hat{\mathbf{c}}_s$, $\hat{\mathbf{A}}_s$, $\hat{\mathbf{E}}_s$, $\hat{\mathbf{c}}_f$, $\hat{\mathbf{A}}_f$, $\hat{\mathbf{E}}_f$, and $\hat{\mathbf{H}}_f$, analogously to the form of the independent regularization parameters in (9), but it would be computationally challenging to determine optimal values for the resulting seven regularization parameters. To mitigate this complexity, in our problem setup we choose to regularize the operators $\hat{\mathbf{c}}_s$, $\hat{\mathbf{A}}_s$ and $\hat{\mathbf{E}}_s$ together so that the structural subproblem's regularization has the form

$$\mathcal{R}_s(\hat{\mathbf{O}}_s) = \gamma_s^{\text{linear}} \left(\|\hat{\mathbf{c}}_s\|_F^2 + \|\hat{\mathbf{A}}_s\|_F^2 + \|\hat{\mathbf{E}}_s\|_F^2 \right)$$

and we choose to regularize the operators $\hat{\mathbf{c}}_f$, $\hat{\mathbf{A}}_f$ and $\hat{\mathbf{E}}_f$ together, so that the fluid subproblem's regularization has the

form

$$\mathcal{R}_f(\widehat{\mathbf{O}}_f) = \gamma_f^{\text{linear}} \left(\|\widehat{\mathbf{C}}_f\|_F^2 + \|\widehat{\mathbf{A}}_f\|_F^2 + \|\widehat{\mathbf{E}}_f\|_F^2 \right) + \gamma_f^{\text{quadratic}} \|\widehat{\mathbf{H}}_f\|_F^2.$$

This choice leads to a three-dimensional regularization hyperparameter space, where we need to select the three regularization parameters γ_s^{linear} , γ_f^{linear} , and $\gamma_f^{\text{quadratic}}$. We perform a coarse, logarithmically-spaced grid search followed by a more targeted, linearly-spaced grid search to identify the optimal regularization values for the three hyperparameters. During these grid searches, we also incorporate the previously mentioned bounded growth constraint, which we specify to be $\alpha = 10$ after a linear grid search.

E. Performance of block-structured Operator Inference

This section compares the performance of the block-structured and monolithic Operator Inference ROMs for the AGARD wing by using the full-order model's predictions and performance as a reference. Section IV.E.1 investigates the accuracy of each ROM type, and then Section IV.E.2 investigates the computational complexity and associated prediction costs of each ROM type.

1. Accuracy

After identifying the appropriate regularization levels for each flow condition, we proceed to analyzing the results of the block-structured Operator Inference ROM. Our quantities of interest for the AGARD wing are the lift coefficient C_L for the fluid dynamics and the generalized displacements gdisp_i for the structural dynamics. We choose these quantities because the lift coefficient represents an integrated effect of the fluid dynamics on the structural dynamics and the generalized displacements can be used to assess the stability of the coupled dynamics for flutter boundary characterization [49]. We note that the aerodynamic accuracy of the ROMs could be further assessed by analyzing generalized aerodynamic forces. Such an analysis could give additional insight into the conditions where ROM predictions deviate from the corresponding high-fidelity CFD result.

We use the ROMs to predict forward in time to the end of the testing regime ($t_f = 0.2453$, i.e., 1000 timesteps) and compare to the full-order model predictions for each flow condition. For each ROM (learned separately for each of the nine flow conditions), we perturb the aeroelastic system in the same manner as during the training data generation in Sec. IV.A by setting all four generalized velocities to a nonzero initial value of $\text{gvel}_i = 0.1$ and then investigating the dynamics that occur due to the initial perturbations. Figs. 6-11 show the improved accuracy of the block-structured Operator Inference predictions compared to monolithic Operator Inference for the lift coefficient, C_L , and the first two generalized displacements, gdisp_1 and gdisp_2 for $r_f = 8, 12$. We see that the $M_\infty = 0.901$, $q_\infty = 50, 70$ flow conditions are accurately predicted by both methods, which is not surprising because these are the least complex flow conditions. The higher q_∞ flow conditions, which approach the flutter boundary, are more clearly improved by the block-structured method for some quantities, but not for all of them. The supersonic $M_\infty = 1.141$ flow conditions are not perfectly

predicted by either method, but are clearly more stable when using the block-structured method.

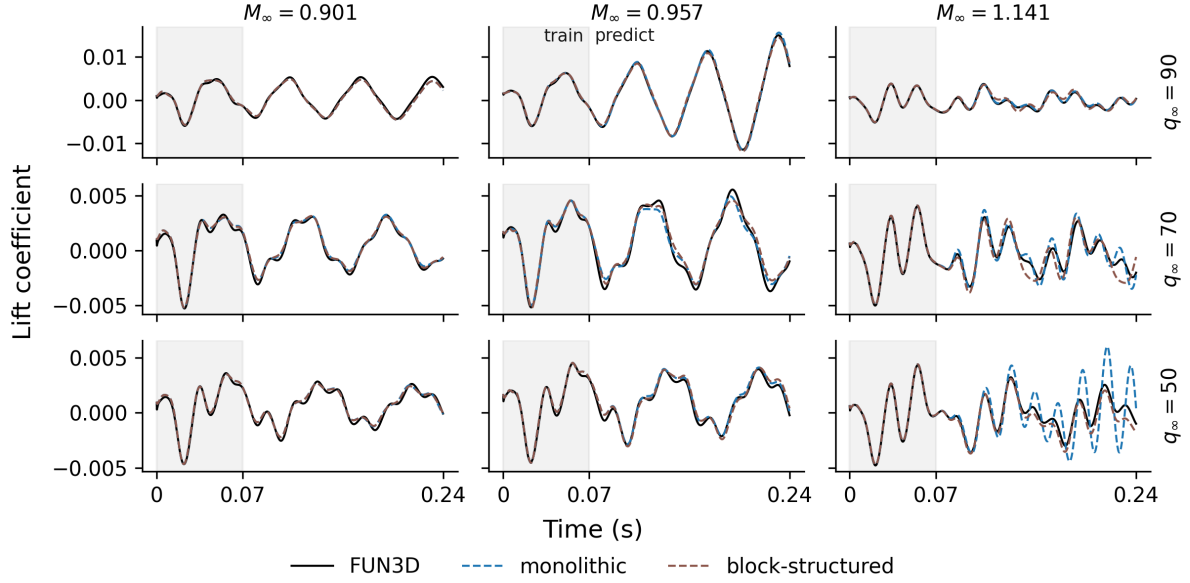


Fig. 6 Lift coefficient (C_L) predictions for monolithic and block-structured Operator Inference with $k_{\text{train}} = 300$ and $r_f = 8$.

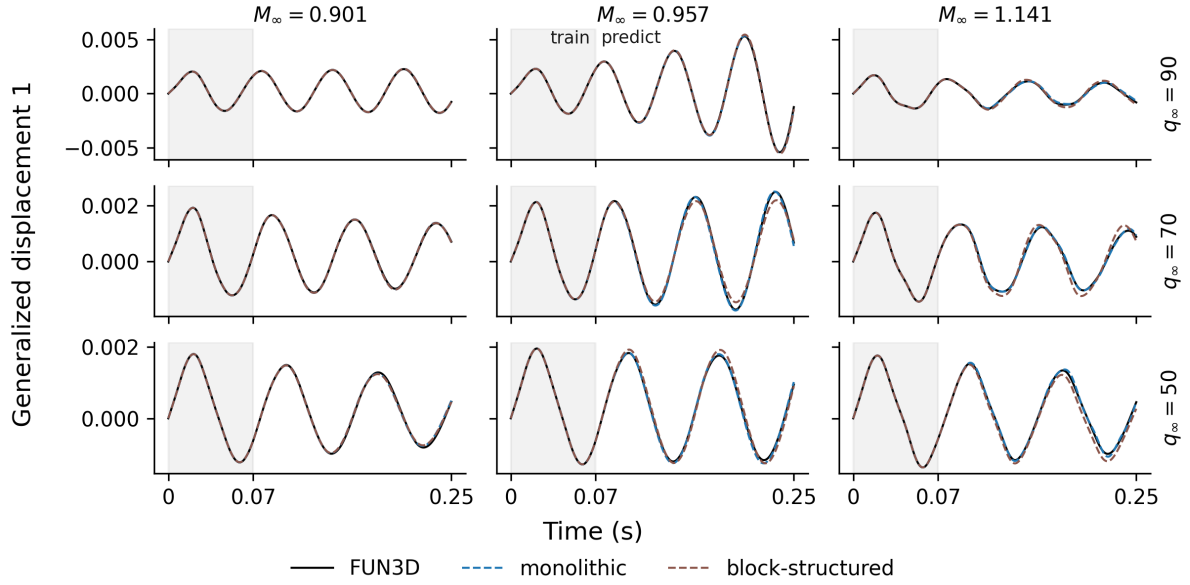


Fig. 7 First generalized displacement ($gdis_1$) predictions for monolithic and block-structured Operator Inference with $k_{\text{train}} = 300$ and $r_f = 8$.

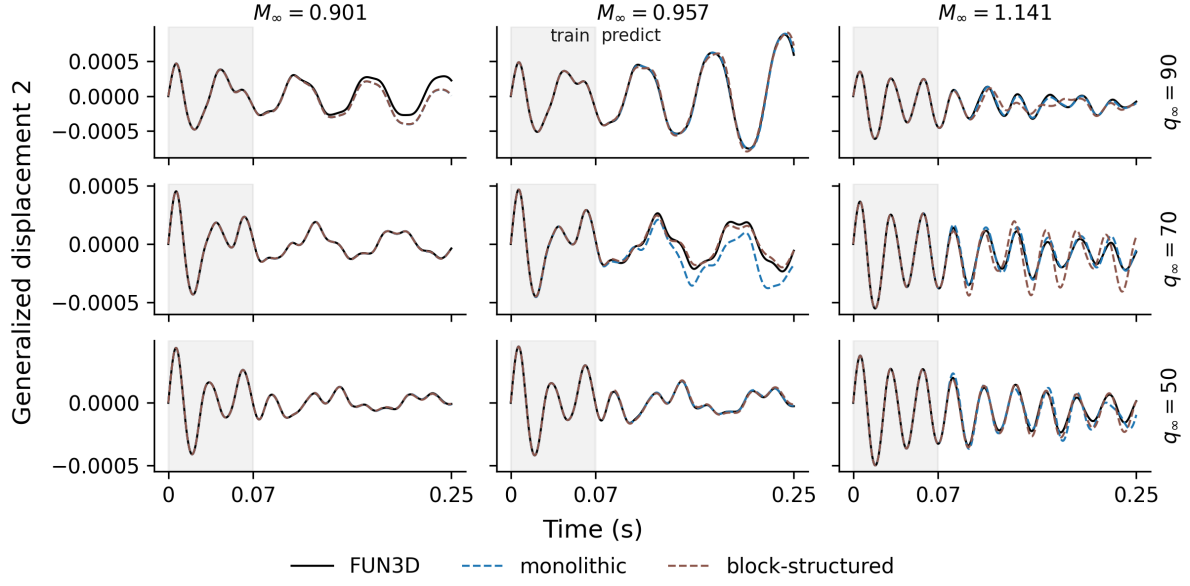


Fig. 8 Second generalized displacement ($gdisp_2$) predictions for monolithic and block-structured Operator Inference with $k_{\text{train}} = 300$ and $r_f = 8$.

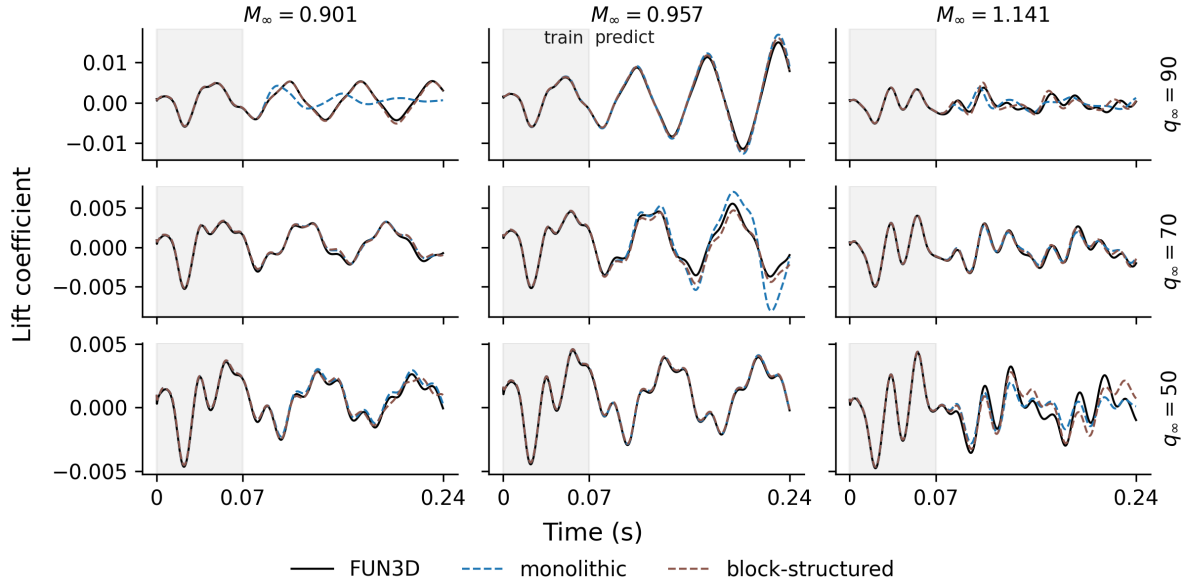


Fig. 9 Lift coefficient (C_L) predictions for monolithic and block-structured Operator Inference with $k_{\text{train}} = 300$ and $r_f = 12$.

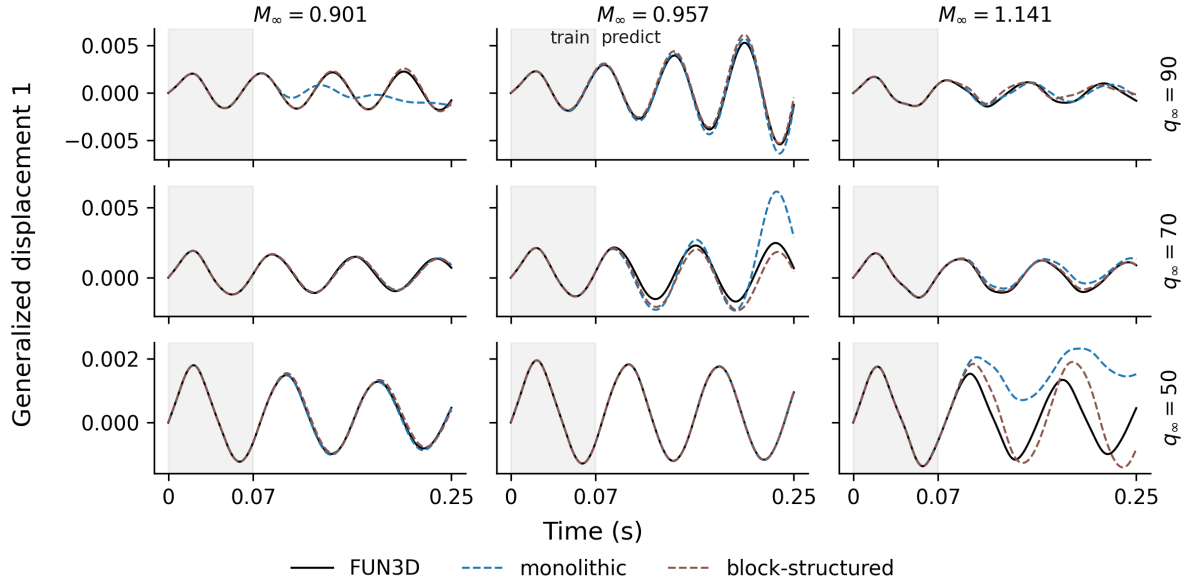


Fig. 10 First generalized displacement ($gdisp_1$) predictions for monolithic and block-structured Operator Inference with $k_{\text{train}} = 300$ and $r_f = 12$.

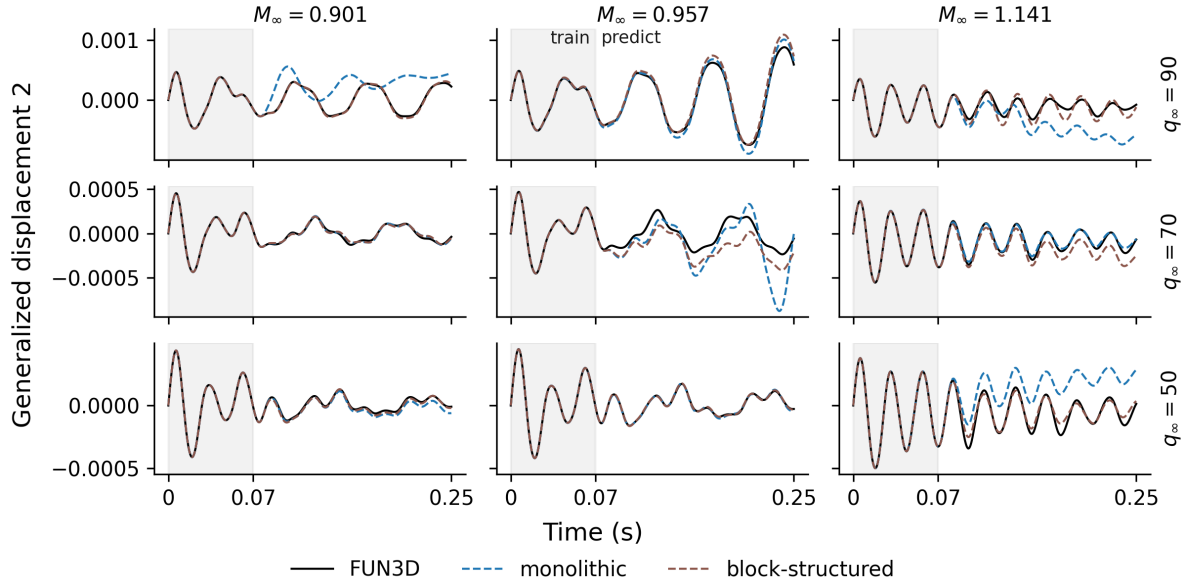


Fig. 11 Second generalized displacement ($gdisp_2$) predictions for monolithic and block-structured Operator Inference with $k_{\text{train}} = 300$ and $r_f = 12$.

The accuracy of pressure distribution prediction on the surface of the wing provides more insight into the ROM's performance, especially because the surface pressure distribution is the source of the lift coefficient calculation. Figures 12 and 13 illustrate the surface pressure distribution and absolute pressure error on the surface of the AGARD wing at three different timesteps corresponding to the end of the training period, $t_{\text{train}} = t(k_{\text{train}})$, and at two and three times the training period, $2t_{\text{train}}$ and $3t_{\text{train}}$. The magnitude of these errors is small due to the small perturbations that we use to excite the dynamics of the system. However, for computations like flutter prediction this is acceptable since we are primarily interested in the stability of the coupled dynamics. We can see that the largest pressure prediction differences occur across the location where the main shock discontinuity is beginning to form. Additionally, the largest errors are along the leading edge of the wing and just upstream and downstream of the shock discontinuity. POD modes are known to struggle with representing discontinuous changes in the state, so this is not surprising.

We compare these error contours to the C_L time histories in Figs. 6 and 9. We see that the poor performance of the monolithic ROM for $r_f = 12$ in the contour plots is because the monolithic C_L approximation does not track the full-order model C_L for that flow condition. Conversely, we see that both ROMs have small C_L errors for $r_f = 8$ (Fig. 6), and the fact that the block-structured ROM has larger error is a function of the specific timesteps at which we plotted the contours.

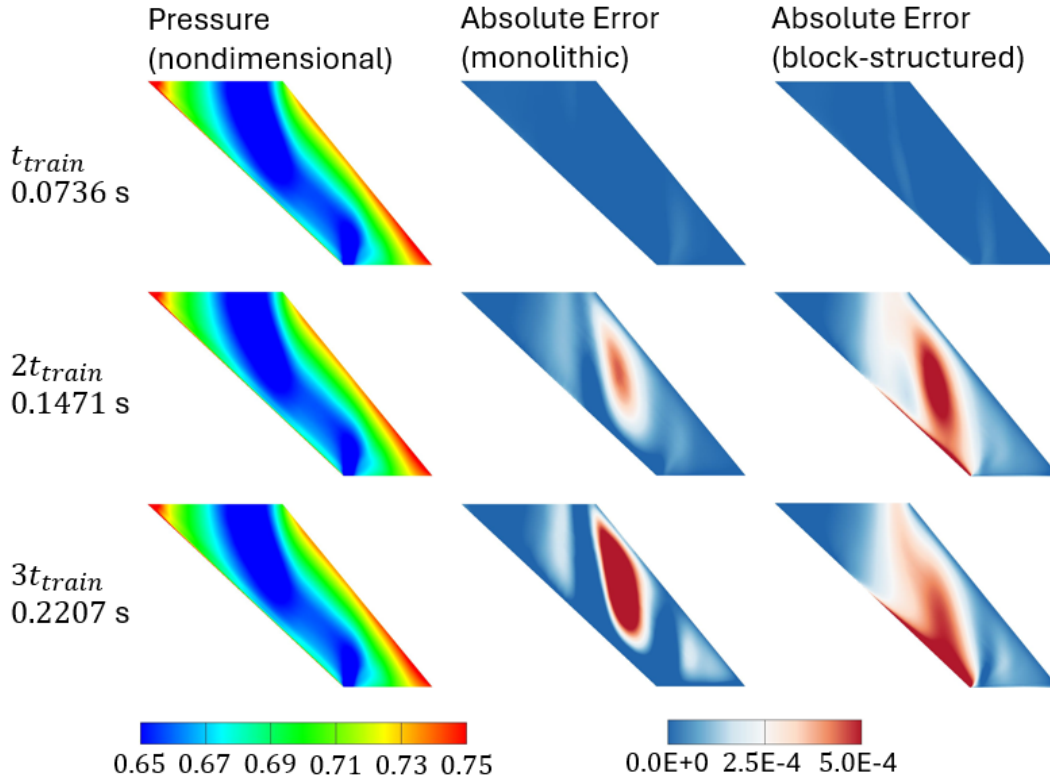


Fig. 12 Pressure and pressure absolute error wing surface contours for $M_\infty = 0.957$, $q_\infty = 70$ psf, $k_{\text{train}} = 300$, and $r_f = 8$.

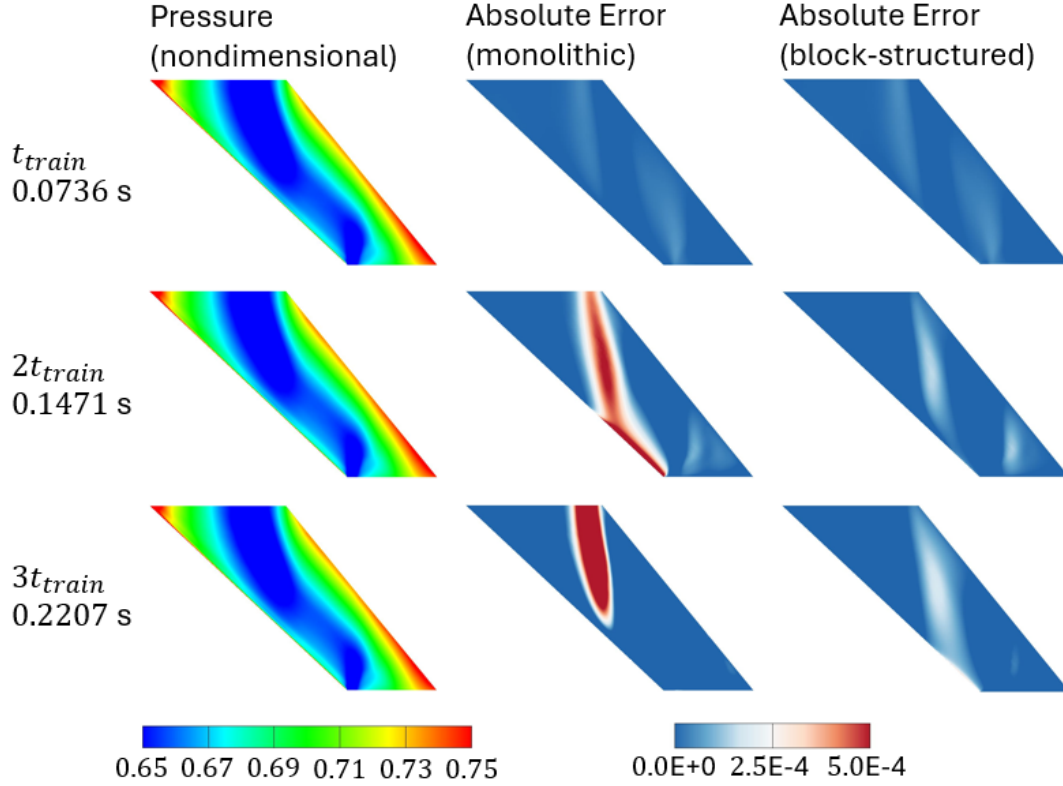


Fig. 13 Pressure and pressure absolute error wing surface contours for $M_\infty = 0.957$, $q_\infty = 70$ psf, $k_{\text{train}} = 300$, and $r_f = 12$.

Inspecting Figs. 6 through 11 reveals tradeoffs between accurately predicting the C_L response and the generalized displacement responses, along with related tradeoffs in accuracy from using different reduced fluid dimensions ($r_f = 8, 12$). We can see that C_L and $gdisp_1$ are predicted more accurately for $r_f = 8$, but $gdisp_2$ is predicted more accurately for $r_f = 12$. As is often the case in POD-based ROMs, the lower-dimension ROM ($r_f = 8$) does an excellent job of capturing the response of lower frequency dynamic behavior (i.e., the first generalized displacement), but does not fully resolve the higher frequency behavior of the second generalized displacement. Increasing the dimension of the POD basis guarantees that the training snapshot data can be represented with lower error, but it does not guarantee more accurate ROM predictions, especially for predictions beyond the end of the training period. For example, we see that the $gdisp_1$ predictions for $r_f = 8$ in Fig. 7 are more accurate than the equivalent predictions for $r_f = 12$ in Fig. 10. This is likely because the $r_f = 12$ ROM overfits to the training data when we increase the reduced dimension, which is a common tradeoff that is seen across data-driven modeling. This hypothesis is supported by the better (more stable) predictions of the block-structured ROM compared to the monolithic ROM for $r_f = 12$, recalling that a benefit of the block-structured approach is the reduced data requirement due to the lower number of operator entries that must be learned (see Sec. IV.E.2 for further discussion of this distinction).

To check the accuracy of the ROMs at other r_f levels, we use a relative root-mean-square-error (RMSE) metric

normalized by the range of the full-order model-predicted quantity of interest over the prediction time period. Here our quantity of interest is the lift coefficient C_L that was plotted in Figs. 6 and 9 above.

We define the lift coefficient error metric as

$$\varepsilon_{\text{rel}}(\mathbf{C}_L) = \frac{\sqrt{\frac{1}{k_{\text{predict}}} \sum_{i=1}^{k_{\text{predict}}} (C_{L,i}^{\text{FOM}} - C_{L,i}^{\text{ROM}})^2}}{\max(\mathbf{C}_L^{\text{FOM}}) - \min(\mathbf{C}_L^{\text{FOM}})} \quad (34)$$

where $\mathbf{C}_L^{\text{FOM}} \in \mathbb{R}^{k_{\text{predict}}}$ and $\mathbf{C}_L^{\text{ROM}} \in \mathbb{R}^{k_{\text{predict}}}$ are the full-order model-predicted and ROM-predicted lift coefficients over all timesteps in the prediction regime $[t_{\text{train}}, t_{\text{final}}] = [0.0736, 0.2453]$ seconds and where k_{predict} is the number of timesteps in the prediction regime. Across the flow conditions, C_L ranges from 0.005391 to 0.026499, thus providing a physically meaningful normalization value for each flow condition to assess the magnitude of the ROM prediction errors.

Figure 14 shows the error metric $\varepsilon_{\text{rel}}(\mathbf{C}_L)$ in (34) for the lift coefficient under varying reduced fluid dimension and all nine flow conditions. We see that there is not a clear relationship between increasing reduced fluid dimension and decreasing relative error. Thus we should focus on choosing a reduced fluid dimension such that the ROM is as stable as possible, especially since Fig. 14 illustrates that there are some choices of r_f that lead to more unstable (inaccurate) ROMs. Overall, we see that there is not an especially clear pattern of one method outperforming the other, but the block-structured method tends to match the monolithic Operator Inference performance, with each method having a few poor-performing data points, while the block-structured method uses a significantly smaller ROM (with a faster prediction time), as we discuss in Sec. IV.E.2.

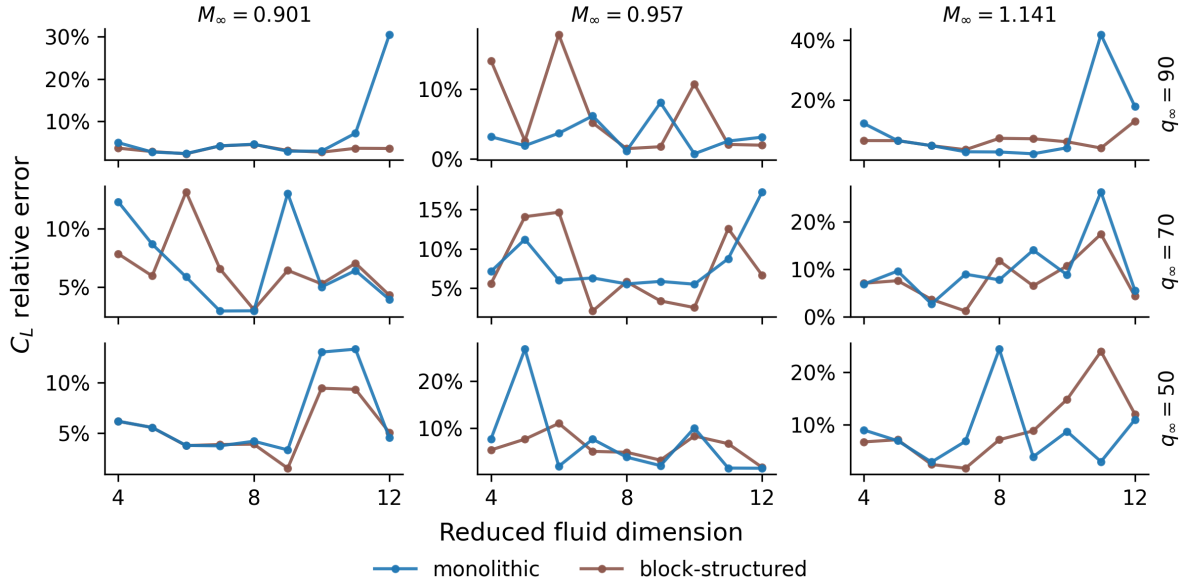


Fig. 14 Time-integrated lift coefficient relative prediction error comparing monolithic and block-structured Operator Inference with $k_{\text{train}} = 300$.

2. Computational complexity

A significant benefit of the block-structured Operator Inference method is the reduced computational expense of the learning and prediction steps. This savings is due to the smaller number of nonzero operator entries that must be learned in the least squares step and computed during each temporal propagation step. Figure 15 illustrates these savings for a constant $r_s = 8$ and varying r_f . We see that for small r_f , the percent reduction in the number of entries to learn is large, but the actual decrease in the number of entries is small. Conversely, for large r_f , the percent reduction gets smaller but the actual decrease in the number of entries grows with the increase in r_f . Of course, this relationship is specific to the block-structure of the aeroelastic problem studied in this paper, and different savings trends would be observed for dynamics with different mathematical structure.

For small reduced fluid dimension (r_f), the difference in computational complexity between the ROMs is mostly due to the unnecessary inclusion of the quadratic structural operator $\hat{\mathbf{H}}_s$ in the monolithic ROM, since the linear, bilinear, and quadratic coupling operators have relatively few terms. As r_f increases, the complexity of the fluid dynamics operators and the coupling operators begins to dominate in both ROMs. However, some of these operators are then set to zero for the block-structured ROM, thus illustrating the reduced complexity of block-structured Operator Inference. This appears in Fig. 15a as the increasing gap between the two lines, leading to the behavior in Fig. 15b. To summarize, the complexity difference is due entirely to the imposition of zero blocks in the block-structured ROM, but the magnitude of the difference is dependent on the reduced fluid dimension, at least for the AGARD example. In other model reduction applications, intrusively known operator blocks or other block sparsity will also lead to reduced complexity.

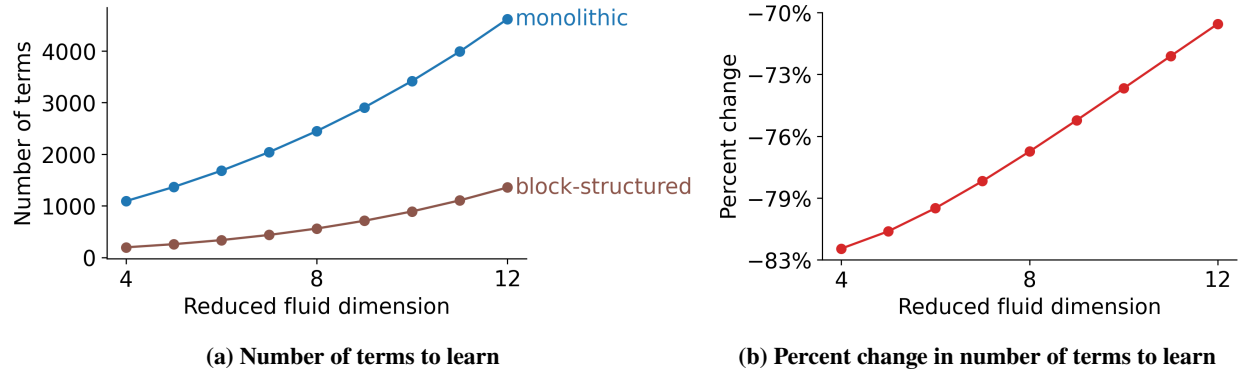


Fig. 15 Computational complexity of the aeroelastic AGARD Operator Inference problem for varying reduced fluid dimension.

To illustrate the practical benefits of having reduced computational complexity, Fig. 16 illustrates the median and 25th to 75th interpercentile range of the online ROM prediction time across the nine flow conditions. For example, we see that at $r_f = 12$, the median prediction time decreases from 0.23 seconds using the monolithic ROM to 0.18 seconds using the block-structured ROM, approximately a 21% decrease. This trend is consistent across all of the studied r_f values (reduced fluid dimensions). Unsurprisingly, some flow conditions have more complicated dynamics and thus

require longer computation times for prediction, but the relative cost between the two ROM methods remains similar across flow conditions, with the block-structured ROM performing 20.6% faster than the monolithic ROM on average across flow conditions and reduced fluid dimensions for $k_{\text{train}} = 300$.

This prediction speedup helps in two notable ways. First, the many-query setting is commonly where ROMs find their most practical engineering applications. For example, uncertainty quantification via Monte Carlo estimation typically requires many thousands of evaluations of a model to obtain a single estimate of a statistical quantity of interest. In such an application, even though the wall-clock time of the gains achieved by block-structured Operator Inference is small for a single run, the percentage gain of 20% is high, and over an entire Monte Carlo simulation it would translate into practically useful wall-clock gains. This is why many-query settings are able to amortize the offline cost of building a ROM by comparing to the online costs of the many predictions required after the ROM is constructed. Second, during the regularization grid search we must infer a new Operator Inference ROM for each possible set of regularization levels and then use that new ROM to make predictions in order to identify the optimal regularization levels. That grid search can be a significant portion of the offline cost of building the ROM, so speedups can enable faster or more thorough grid searches.

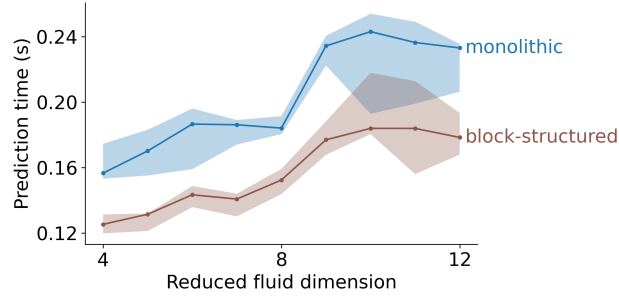


Fig. 16 Prediction times averaged over flow conditions for varying reduced fluid dimension with median (50th percentile) line and 25th–75th interpercentile shaded region.

3. Prediction outside of training set

We now investigate the accuracy of the monolithic and block-structured Operator Inference ROMs for perturbations (initial conditions) that are not included in the training set. We test this capability at the $M_\infty = 0.901$, $q_\infty = 70$ psf flow condition by only perturbing the initial generalized velocity $gvel_i$ for a single structural mode per test case, rather than perturbing all four modes simultaneously. We keep the initial generalized displacements $gdispi$ equal to zero for all four cases, but now in case 1, for example, we set $gvel_1 = 0.1$ while keeping $gvel_2 = gvel_3 = gvel_4 = 0.0$. We repeat this process for each of the four structural modes, resulting in four new test cases with a single perturbation per case. We investigated the ROMs' accuracy for both $r_f = 8$ and $r_f = 12$, but found that both the block-structured and monolithic Operator Inference ROMs made similarly accurate predictions for $r_f = 12$, so we focus on the $r_f = 8$ ROMs

here to highlight the differences.

Figures 17–19 show the ability of the monolithic and block-structured Operator Inference ROMs to accurately predict these new cases that are not present in the training dataset. The lift coefficient predictions in Fig. 17 show that the block-structured ROM is significantly better than the monolithic ROM at replicating the FUN3D-predicted lift coefficient behavior, especially as the simulations progress further in time. In contrast, the monolithic ROM struggles to generalize to the new initial conditions and we see that for all four cases the monolithic ROM incorrectly predicts that the lift coefficient oscillations will unstably increase over time.

Similar trends can be seen in the ROMs’ predictions of the generalized displacements in Figs. 18–19. Both ROMs are able to accurately predict the response of $gdisp_1$ in Fig. 18, but in Fig. 19 the block-structured ROM is able to correctly predict the response of $gdisp_2$ for each of the four cases while the monolithic ROM is much less accurate or even unstable. This may be due to the improved ability of the block-structured approach to precisely target the known governing equations and thus the known physics, while the monolithic ROM experiences overfitting without the additional information embedded through the coupling and block structure.

We also investigated the performance of a linear Operator Inference ROM (i.e., with no quadratic operator for either physics regime) due to the popularity of linear methods and the small magnitude of the perturbations we have imposed. We found that while the linear ROM made accurate predictions for the perturbations within the training set in Sec. IV.E.1, it struggled to make predictions for these new perturbations that were not in the training set. We have omitted these predictions from the figures for clarity, but this investigation gives us additional confidence that our quadratic, block-structured ROM is accurately predicting the nonlinear behavior of the coupled system.

V. Conclusion

Data-driven model reduction is often limited by a lack of physically or mathematically informed structure which can lead to overfitting and unstable dynamics. Therefore, incorporating structure into ROMs is desirable, especially in the coupled multiphysics context where problem structure plays a key role in determining the dynamical behavior of the coupled system. This paper develops the block-structured Operator Inference method to embed structure in the ROM learning problem. Taking advantage of the block structure of a multiphysics system enables separate specification of the governing equation form for each physics regime and tailored regularization choices during the least squares inference step. The application of block-structured Operator Inference to the high-dimensional AGARD 445.6 wing aeroelastic modeling problem demonstrates the ability to learn ROMs that preserve the accuracy of monolithic Operator Inference for initial conditions in the training set, while in many cases improving upon that accuracy for initial conditions outside of the training set. The block-structured ROMs are also shown to reduce the computational expense of online prediction by an average of 20% across flow conditions and reduced dimensions.

While the problem studied in this paper employs linear coupling, the approach admits bilinear and quadratic coupling

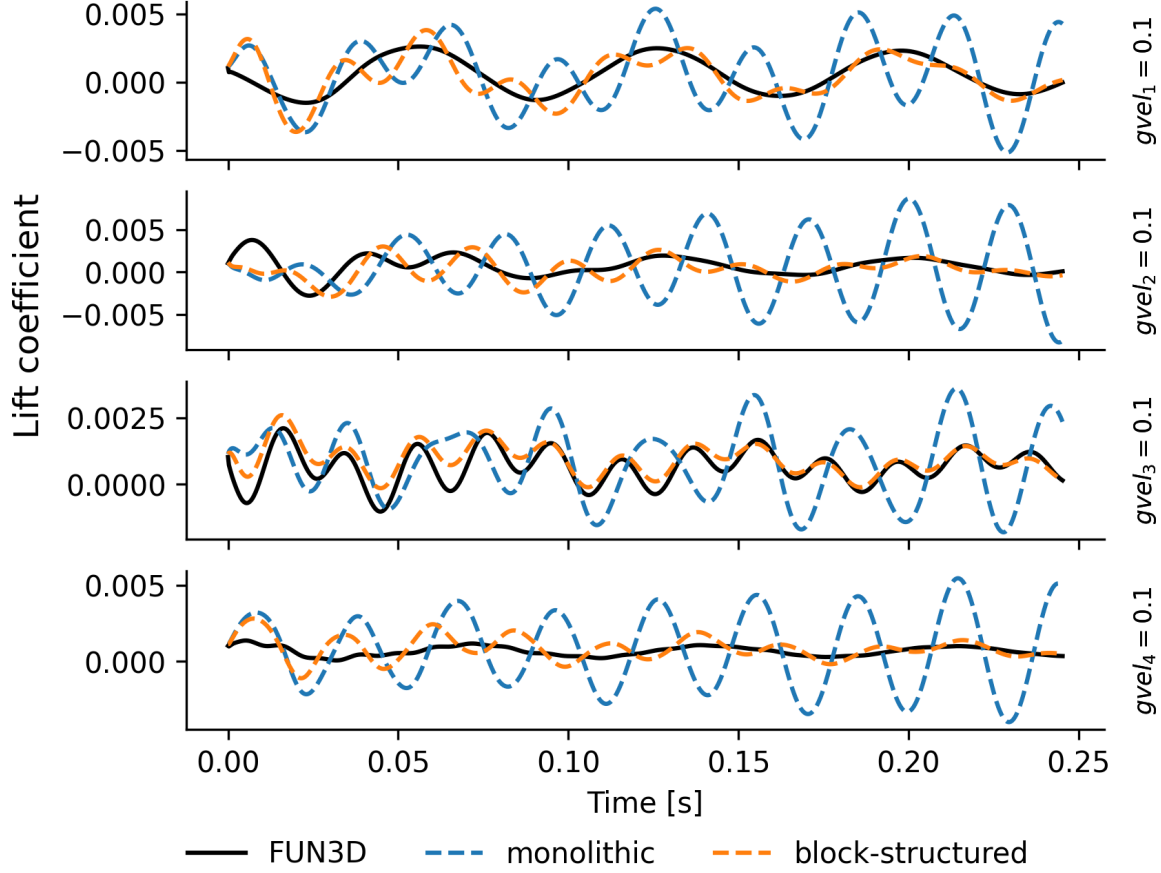


Fig. 17 Lift coefficient predictions for an initial perturbation of $gvel_i = 0.1, i = 1, \dots, 4$ with $k_{\text{train}} = 300$ and $r_f = 8$.

where dictated by the underlying physical problem. The approach also enables the specification of known operator blocks intrusively. An extension of the proposed approach could incorporate parametric ROMs that further exploit the system structure exposed in the block-structured ROM. For example, flutter prediction could be achieved by interpolation of primarily the fluid dynamics operators on a manifold, with the structural dynamics operators being parametrically dependent on the flow condition at most in the constant term, rather than interpolating all of the fluid and structural operators together. Embedding known block structure into nonintrusive ROMs, especially for the high-dimensional multiphysics setting, provides an important step in the direction of increasing the flexibility and expressiveness of the data-driven Operator Inference method.

Appendix: FUN3D implementation of flow condition selection inspired by flutter boundary characterization

Algorithm 1 describes the process for selecting the FUN3D input parameters, listed in Table 1, that are needed to run an AGARD 445.6 wing simulation at a specific flow condition within the flutter characterization parameter space

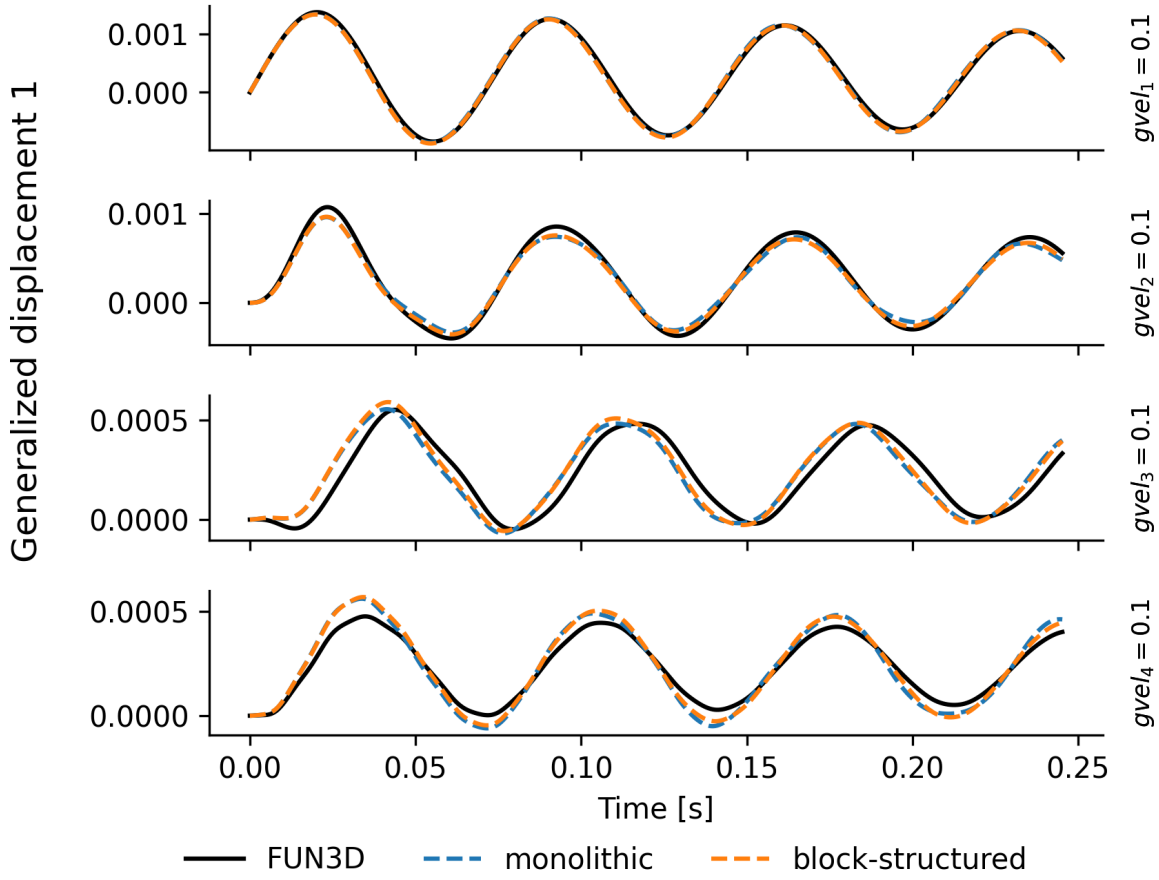


Fig. 18 First generalized displacement predictions for an initial perturbation of $gvel_i = 0.1, i = 1, \dots, 4$ with $k_{\text{train}} = 300$ and $r_f = 8$.

(M_∞, q_∞) . These parameters were chosen to be consistent with the selected freestream mach number $M_\infty \in \mathcal{M}$ and freestream dynamic pressure $q_\infty \in \mathcal{Q}$. We require the mach number M_∞ and dynamic pressure q_∞ for the particular flow condition of interest and a series of constant parameters described in Tab. 2. Table 3 lists the FUN3D names for each of the required input parameters, including those computed by Alg. 1, for ease of identification and entry into the FUN3D input namelist file.

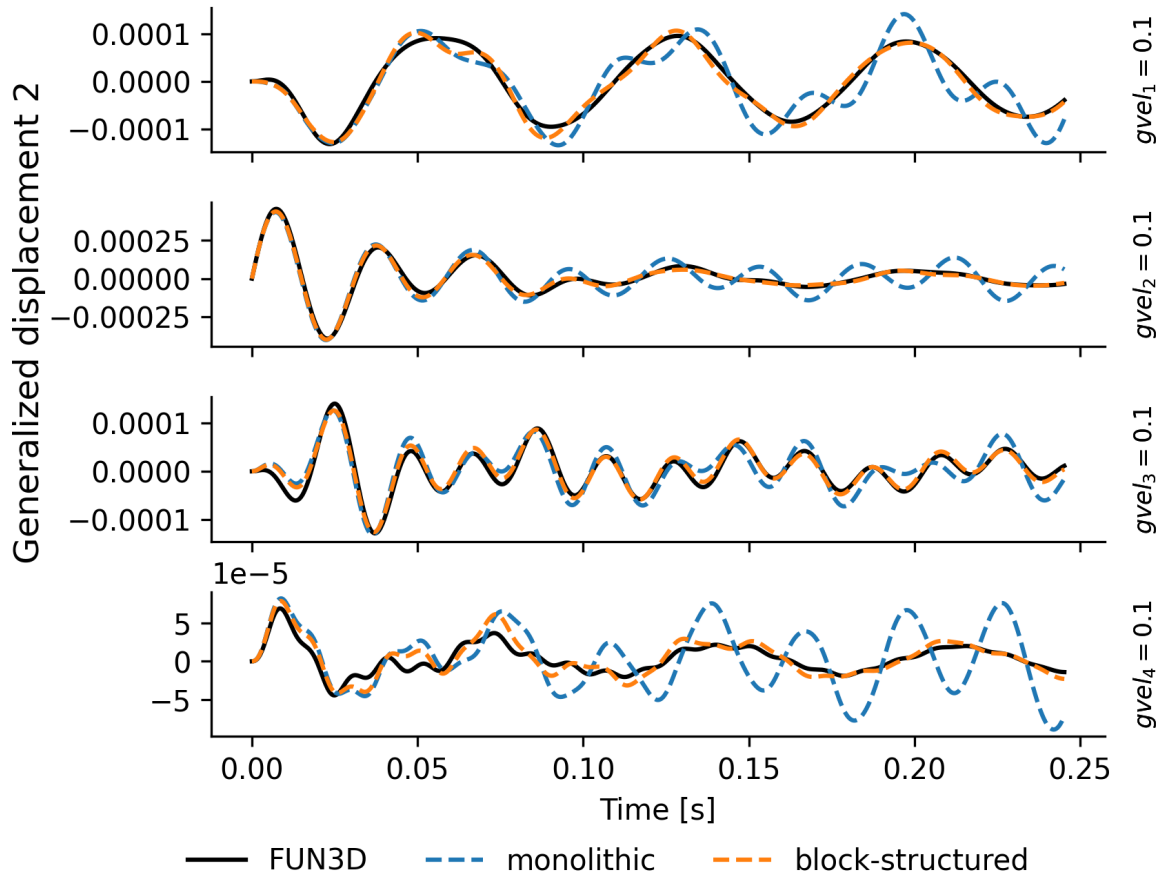


Fig. 19 Second generalized displacement predictions for an initial perturbation of $gvel_i = 0.1, i = 1, \dots, 4$ with $k_{\text{train}} = 300$ and $r_f = 8$.

Algorithm 1 Selection of input parameters for a FUN3D run (originally for flutter characterization)

Input: $M_\infty, q_\infty, L, L_{\text{nondim}}, f_{\text{char}}, N, \gamma, R, C, T_{\text{ref}}, \mu_0$ **Output:** $u_\infty, T, Re_L, \Delta t_{\text{nondim}}$

- 1: $\rho \leftarrow \rho(M_\infty)$ ▷ Fix density at original experimental flutter value in [22]
 - 2: $u_\infty \leftarrow \sqrt{2q_\infty/\rho}$ ▷ Compute freestream velocity
 - 3: $a \leftarrow \frac{u_\infty}{M_\infty}$ ▷ Compute speed of sound
 - 4: $T \leftarrow a^2/\gamma R$ ▷ Compute temperature
 - 5: $\mu \leftarrow \mu_0 \frac{T_{\text{ref}}+C}{T+C} \left(\frac{T}{T_{\text{ref}}} \right)^{3/2}$ ▷ Compute viscosity via Sutherland's Law
 - 6: $Re \leftarrow \frac{\rho u_\infty L}{\mu}$ ▷ Compute Reynolds number
 - 7: $Re_L \leftarrow \frac{Re}{L_{\text{nondim}}}$ ▷ Compute Reynolds number per nondimensional unit length
 - 8: $\Delta t = \frac{1}{f_{\text{char}}}/N$ ▷ Compute dimensional timestep size
 - 9: $\Delta t_{\text{nondim}} \leftarrow a \frac{L_{\text{nondim}}}{L} \Delta t$ ▷ Compute nondimensional timestep size
-

Table 1 Parameter values corresponding to each flow condition in the training data generation set. Note that the time step size is nondimensional.

Mach number	Dynamic pressure	Freestream Velocity	Temperature	Reynolds number	Time step size
M_∞ [-]	q_∞ [lb/ft ²]	u_∞ [ft/s]	T [°R]	Re [-]	Δt [-]
0.901	50	728.37	271.95	1.164e6	0.19826
0.901	70	861.82	380.72	1.023e6	0.23458
0.901	90	877.21	489.50	9.453e5	0.26599
0.957	50	916.65	381.78	6.856e5	0.23491
0.957	70	1084.60	534.49	6.186e5	0.27794
0.957	90	1229.82	687.21	5.813e5	0.31516
1.141	50	823.66	216.85	1.276e6	0.17704
1.141	70	974.57	303.59	1.102e6	0.20947
1.141	90	1105.06	390.33	1.005e6	0.23752

Table 2 Constants used for computing flutter flow conditions for FUN3D input in Alg. 1

Variable	Symbol	Value	Units	Assumptions
Characteristic length	L	1.833	ft	AGARD wing root chord
Nondimensional characteristic length	L_{nondim}	1.833	-	-
Frequency of interest	f_{char}	20.39	Hz	Highest flutter frequency in [22]
Number of timesteps per period	N	200	-	-
Specific heat ratio	γ	1.4	-	Air, ideal gas
Specific gas constant	R	1716.49	ft-lbf/slug-°R	Air, ideal gas
Sutherland's constant	C	198.6	°R	Air
Reference temperature	T_{ref}	518.69	°R	Standard atmosphere, sea level
Reference viscosity	μ_0	3.737e-7	slug/ft-s	Standard atmosphere, sea level

Table 3 FUN3D input parameter names

Variable	FUN3D parameter name
u_∞	uinf
q_∞	qinf
T	temperature
Re_L	reynolds_number
M_∞	mach_number
Δt_{nondim}	time_step_nondim

Funding Sources

This work was supported in part by Lockheed Martin University Research Agreement MRA-16-005-RPP012-001, AFOSR grant FA9550-21-1-0084, and DOE grant DE-SC002317.

Acknowledgments

We thank Kevin Jacobson and Pawel Chwalowski of the NASA Langley Research Center for their advice and insights regarding the setup of the AGARD 445.6 Wing model in FUN3D and for providing the AGARD 445.6 wing CFD grid.

References

- [1] Sirovich, L., “Turbulence and the dynamics of coherent structures. I. Coherent structures,” *Quarterly of Applied Mathematics*, Vol. 45, No. 3, 1987, pp. 561–571.
- [2] Berkooz, G., Holmes, P., and Lumley, J. L., “The proper orthogonal decomposition in the analysis of turbulent flows,” *Annual Review of Fluid Mechanics*, Vol. 25, 1993, pp. 539–575.
- [3] Lumley, J. L., “The Structures of Inhomogeneous Turbulent Flow,” *Atmospheric Turbulence and Radio Wave Propagation*, 1967, pp. 166–178.
- [4] Rathinam, M., and Petzold, L., “A New Look at Proper Orthogonal Decomposition,” *SIAM Journal on Numerical Analysis*, Vol. 41, No. 5, 2003, pp. 1893–1925. <https://doi.org/10.1137/S0036142901389049>.
- [5] Benner, P., Güğercin, S., and Willcox, K., “A survey of projection-based model reduction methods for parametric dynamical systems,” *SIAM Review*, Vol. 57, No. 4, 2015, pp. 483–531. <https://doi.org/10.1137/130932715>.
- [6] Antoulas, A. C., *Approximation of Large-Scale Dynamical Systems*, SIAM, Philadelphia, PA, 2005.
- [7] Schmid, P. J., “Dynamic mode decomposition of numerical and experimental data,” *Journal of Fluid Mechanics*, Vol. 656, 2010, p. 5–28. <https://doi.org/10.1017/S0022112010001217>.
- [8] Tu, J. H., Rowley, C. W., Luchtenburg, D. M., Brunton, S. L., and Kutz, J. N., “On dynamic mode decomposition: Theory and applications,” *Journal of Computational Dynamics*, Vol. 1, No. 2, 2014, pp. 391–421. <https://doi.org/10.3934/jcd.2014.1.391>.

- [9] Hurty, W. C., “Vibrations of Structural Systems by Component Mode Synthesis,” *Journal of the Engineering Mechanics Division*, Vol. 86, No. 4, 1960, pp. 51–69. <https://doi.org/10.1061/JMCEA3.0000162>, URL <https://ascelibrary.org/doi/abs/10.1061/JMCEA3.0000162>.
- [10] Craig, R. R., and Bampton, M. C. C., “Coupling of substructures for dynamic analyses,” *AIAA Journal*, Vol. 6, No. 7, 1968, pp. 1313–1319. <https://doi.org/10.2514/3.4741>, URL <https://doi.org/10.2514/3.4741>.
- [11] Reis, T., and Stykel, T., *A Survey on Model Reduction of Coupled Systems*, Springer, Berlin, Heidelberg, 2008, Chap. A Survey on Model Reduction of Coupled Systems, pp. 133–155. https://doi.org/10.1007/978-3-540-78841-6_7, URL https://doi.org/10.1007/978-3-540-78841-6_7.
- [12] Benner, P., and Feng, L., “Model order reduction for coupled problems,” *Appl Comput Math Int J*, Vol. 14, 2015, pp. 3–22.
- [13] Discacciati, N., and Hesthaven, J. S., “Model reduction of coupled systems based on non-intrusive approximations of the boundary response maps,” *Computer Methods in Applied Mechanics and Engineering*, Vol. 420, 2024, p. 116770. <https://doi.org/https://doi.org/10.1016/j.cma.2024.116770>, URL <https://www.sciencedirect.com/science/article/pii/S0045782524000264>.
- [14] Gkimitis, L., Richter, T., and Benner, P., “Adjacency-based, non-intrusive reduced-order modeling for fluid-structure interactions,” *Proceedings in Applied Mathematics and Mechanics*, Vol. 23, No. 4, 2023, p. e202300047. <https://doi.org/https://doi.org/10.1002/pamm.202300047>, URL <https://onlinelibrary.wiley.com/doi/abs/10.1002/pamm.202300047>.
- [15] Lall, S., Krysl, P., and Marsden, J. E., “Structure-preserving model reduction for mechanical systems,” *Physica D: Nonlinear Phenomena*, Vol. 184, No. 1-4, 2003, pp. 304–318.
- [16] Beattie, C., and Gugercin, S., “Interpolatory projection methods for structure-preserving model reduction,” *Systems & Control Letters*, Vol. 58, No. 3, 2009, pp. 225–232.
- [17] Gugercin, S., Polyuga, R. V., Beattie, C., and Van Der Schaft, A., “Structure-preserving tangential interpolation for model reduction of port-Hamiltonian systems,” *Automatica*, Vol. 48, No. 9, 2012, pp. 1963–1974.
- [18] Peherstorfer, B., and Willcox, K., “Data-driven Operator Inference for Nonintrusive Projection-Based Model Reduction,” *Computer Methods in Applied Mechanics and Engineering*, Vol. 306, 2016, pp. 196–215. <https://doi.org/10.1016/j.cma.2016.03.025>.
- [19] Kramer, B., Peherstorfer, B., and Willcox, K. E., “Learning nonlinear reduced models from data with operator inference,” *Annual Review of Fluid Mechanics*, Vol. 56, No. 1, 2024, pp. 521–548.
- [20] Sharma, H., Najera-Flores, D. A., Todd, M. D., and Kramer, B., “Lagrangian operator inference enhanced with structure-preserving machine learning for nonintrusive model reduction of mechanical systems,” *Computer Methods in Applied Mechanics and Engineering*, Vol. 423, 2024, p. 116865. <https://doi.org/https://doi.org/10.1016/j.cma.2024.116865>, URL <https://www.sciencedirect.com/science/article/pii/S004578252400121X>.

- [21] Benner, P., Goyal, P., Kramer, B., Peherstorfer, B., and Willcox, K., “Operator inference for non-intrusive model reduction of systems with non-polynomial nonlinear terms,” *Computer Methods in Applied Mechanics and Engineering*, Vol. 372, 2020, p. 113433.
- [22] Yates, E. C. J., “AGARD Standard Aeroelastic Configurations for Dynamic Response. Candidate Configuration I.-Wing 445.6,” Tech. rep., NASA Langley Research Center, Hampton, Virginia, 08 1987.
- [23] Silva, W. A., “Simultaneous Excitation of Multiple-Input/Multiple-Output CFD-Based Unsteady Aerodynamic Systems,” *Journal of Aircraft*, Vol. 45, No. 4, 2008, pp. 1267–1274. <https://doi.org/10.2514/1.34328>.
- [24] Juang, J.-N., and Pappa, R. S., “An Eigensystem Realization Algorithm for Modal Parameter Identification and Model Reduction,” *Journal of Guidance, Control, and Dynamics*, Vol. 8, No. 5, 1985, pp. 620–627. <https://doi.org/10.2514/3.20031>.
- [25] Silva, W., “Identification of Nonlinear Aeroelastic Systems Based on the Volterra Theory: Progress and Opportunities,” *Nonlinear Dynamics*, Vol. 39, No. 1, 2005, pp. 25–62. <https://doi.org/10.1007/s11071-005-1907-z>, URL <https://doi.org/10.1007/s11071-005-1907-z>.
- [26] Silva, W., and Bartels, R., “Development of reduced-order models for aeroelastic analysis and flutter prediction using the CFL3Dv6.0 code,” *Journal of Fluids and Structures*, Vol. 19, No. 6, 2004, pp. 729–745. <https://doi.org/https://doi.org/10.1016/j.jfluidstructs.2004.03.004>, URL <https://www.sciencedirect.com/science/article/pii/S088997460400057X>.
- [27] Silva, W. A., “AEROM: NASA’s Unsteady Aerodynamic and Aeroelastic Reduced-Order Modeling Software,” *Aerospace*, Vol. 5, No. 2, 2018. <https://doi.org/10.3390/aerospace5020041>.
- [28] Lowe, B. M., and Zingg, D. W., “Efficient Flutter Prediction Using Reduced-Order Modeling,” *AIAA Journal*, Vol. 59, No. 7, 2021, pp. 2670–2683. <https://doi.org/10.2514/1.J060006>.
- [29] Hall, K. C., Thomas, J. P., and Dowell, E. H., “Proper Orthogonal Decomposition Technique for Transonic Unsteady Aerodynamic Flows,” *AIAA Journal*, Vol. 38, No. 10, 2000, pp. 1853–1862. <https://doi.org/10.2514/2.867>, URL <https://doi.org/10.2514/2.867>.
- [30] Dowell, E. H., and Hall, K. C., “Modeling of Fluid-Structure Interaction,” *Annual Review of Fluid Mechanics*, Vol. 33, 2001, pp. 445–490. <https://doi.org/10.1146/annurev.fluid.33.1.445>.
- [31] Lieu, T., Farhat, C., and Lesoinne, M., “Reduced-order fluid/structure modeling of a complete aircraft configuration,” *Computer Methods in Applied Mechanics and Engineering*, Vol. 195, No. 41–43, 2006, pp. 5730–5742. <https://doi.org/10.1016/j.cma.2005.08.026>.
- [32] Lieu, T., and Farhat, C., “Adaptation of Aeroelastic Reduced-Order Models and Application to an F-16 Configuration,” *AIAA Journal*, Vol. 45, No. 6, 2007, pp. 1244–1257. <https://doi.org/10.2514/1.24512>.
- [33] Amsallem, D., and Farhat, C., “Interpolation Method for the Adaptation of Reduced-Order Models to Parameter Changes and Application to Aeroelasticity,” *AIAA Journal*, Vol. 46, No. 7, 2008, pp. 1803–1813. <https://doi.org/10.2514/1.35374>.

- [34] Hall, K. C., “Eigenanalysis of Unsteady Flows about Airfoils, Cascades, and Wings,” *AIAA Journal*, Vol. 32, No. 12, 1994, pp. 2426–2432. <https://doi.org/10.2514/3.12309>.
- [35] Bouhlel, M. A., Bartoli, N., Otsmane, A., and Morlier, J., “Improving kriging surrogates of high-dimensional design models by Partial Least Squares dimension reduction,” *Struct Multidisc Optim*, Vol. 53, 2016, pp. 935–952. <https://doi.org/10.1007/s00158-015-1395-9>.
- [36] Kim, T., “Frequency-Domain Karhunen-Loeve Method and Its Application to Linear Dynamic Systems,” *AIAA Journal*, Vol. 36, No. 11, 1998, pp. 2117–2123. <https://doi.org/10.2514/2.315>, URL <https://doi.org/10.2514/2.315>.
- [37] Biedron, R. T., Carlson, J.-R., Derlaga, J. M., Gnoffo, P. A., Hammond, D. P., Jones, W. T., Kleb, B., Lee-Rausch, E. M., Nielsen, E. J., Park, M. A., Rumsey, C. L., Thomas, J. L., Thompson, K. B., and Wood, W. A., *FUN3D Manual 13.6*, NASA Langley Research Center, Hampton, Virginia, 2019.
- [38] Kramer, B., and Willcox, K. E., “Nonlinear model order reduction via lifting transformations and proper orthogonal decomposition,” *AIAA Journal*, Vol. 57, No. 6, 2019, pp. 2297–2307.
- [39] Qian, E., Kramer, B., Peherstorfer, B., and Willcox, K., “Lift & Learn: Physics-informed machine learning for large-scale nonlinear dynamical systems,” *Physica D: Nonlinear Phenomena*, Vol. 406, 2020, p. 132401. <https://doi.org/https://doi.org/10.1016/j.physd.2020.132401>.
- [40] McQuarrie, S. A., Huang, C., and Willcox, K. E., “Data-driven reduced-order models via regularised Operator Inference for a single-injector combustion process,” *Journal of the Royal Society of New Zealand*, Vol. 51, No. 2, 2021, pp. 194–211. <https://doi.org/10.1080/03036758.2020.1863237>.
- [41] Qian, E., Farcas, I., and Willcox, K., “Reduced Operator Inference For Nonlinear Partial Differential Equations,” *SIAM Journal on Scientific Computing*, Vol. 44, 2022, pp. A1934–A1959.
- [42] Schilders, W. H. A., and Lutowska, A., *A Novel Approach to Model Order Reduction for Coupled Multiphysics Problems*, Springer International Publishing, Cham Heidelberg New York Dordrecht London, 2014, Chap. A Novel Approach to Model Order Reduction for Coupled Multiphysics Problems, pp. 1–49. https://doi.org/10.1007/978-3-319-02090-7_1, URL https://doi.org/10.1007/978-3-319-02090-7_1.
- [43] Biedron, R., and Thomas, J., “Recent enhancements to the FUN3D flow solver for moving-mesh applications,” *47th AIAA Aerospace Sciences Meeting including The New Horizons Forum and Aerospace Exposition*, 2009, p. 1360. <https://doi.org/10.2514/6.2009-1360>.
- [44] Chen, R.-H., *Universal Physical Constants*, Cambridge University Press, New York, 2017, Chaps. Appendix A - Universal Physical Constants, p. 357. <https://doi.org/10.1017/9781316014288>, URL <https://doi.org/10.1017/9781316014288>.
- [45] Etkin, B., and Reid, L. D., *Dynamics of Flight: Stability and Control*, Wiley, Hoboken, NJ, 1996.

- [46] Spalart, P., and Allmaras, S. R., “A one-equation turbulence model for aerodynamic flows,” *Le Recherche Aerospatiale*, , No. 1, 1994, pp. 5–21.
- [47] Silva, W. A., Chwalowski, P., and Perry III, B., “Evaluation of linear, inviscid, viscous, and reduced-order modelling aeroelastic solutions of the AGARD 445.6 wing using root locus analysis,” *International Journal of Computational Fluid Dynamics*, Vol. 28, 2014, pp. 122–139. <https://doi.org/10.1080/10618562.2014.922179>.
- [48] Swischuk, R., Kramer, B., Huang, C., and Willcox, K., “Learning Physics-Based Reduced-Order Models for a Single-Injector Combustion Process,” *AIAA Journal*, Vol. 58, No. 6, 2020, pp. 2658–2672. <https://doi.org/10.2514/1.J058943>.
- [49] Jacobson, K. E., Kiviaho, J. F., Kennedy, G. J., and Smith, M. J., “Evaluation of time-domain damping identification methods for flutter-constrained optimization,” *Journal of Fluids and Structures*, Vol. 87, 2019, pp. 174–188.