SAND2020-5692C

Quantifying the impact of the initial guess for projection-based model reduction of steady hypersonic aerodynamics

David S. Ching* and Patrick J. Blonigan†
Sandia National Laboratories, Livermore, California, 94550

Francesco Rizzi
NexGen Analytics, Sheridan, WY, 82801, USA

Micah Howard‡ and Jeffrey A. Fike§
Sandia National Laboratories, Albuquerque, New Mexico, 87185-0828

Hypersonic aerodynamic simulations require extensive computational resources, hindering their usage in hypersonic vehicle design and uncertainty quantification. Projection-based reduced-order models (ROMs) are a computationally inexpensive alternative to full-order simulations that provide major speedup with marginal loss of accuracy. However, ROMs can present robustness and convergence issues, and one source of these issues is the initial guess for the nonlinear solver. This work addresses the effect that the initial guess has on ROM robustness by testing various initial guess interpolation methods and developing a more accurate initial guess method. An accurate initial guess is found to result in significantly faster ROM convergence and higher robustness. In addition, it is found that certain output quantities can be more accurately predicted using an accurate interpolation method than with steady projection-based ROMs for hypersonic flows. It is very important that studies examining steady projection-based ROMs demonstrate improvements in comparison to accurate initial guesses.

I. Introduction

Hypersonic aerodynamics simulations are important for designing reentry vehicles, missiles, and launch vehicles. It is critical that designers are able to accurately predict forces and heating while in the design stage without conducting expensive, time-consuming tests. Because of this, there is an increasing reliance on computational models for design and analysis of hypersonic configurations [1].

One the main challenges associated with hypersonic aerodyanics simulations is their large computational cost: due to the many disparate scales, typical simulation grids must be highly refined close to the body and near shock locations, leading to a very large state space. This makes it computationally expensive to even tackle simplified models such as Reynolds-Averaged Navier-Stokes (RANS) simulations. In addition, to design hypersonic vehicles, engineers need to iterate through hundreds or thousands of designs, each requiring simulations at a variety of speeds, air densities, air temperatures, angle of attack, and airfoil configurations [2], and/or varying flight conditions, vehicle geometry deformation, turbulence model parameters, and boundary layer transition location. These can be classified as *many-query problems*, because they require a large number of model evaluations with different parameters $\boldsymbol{\mu}$. It can be computationally demanding or prohibitively expensive to run full-order models across the parameter space to identify optimal designs and uncertainty estimates. Therefore, there is a strong need for simplified models that run within a reasonable time frame while maintaining high levels of accuracy.

There has been strong interest in projection-based reduced-order modeling (ROM) as a computationally inexpensive alternative to the full-order models (FOMs) for *many-query problems* [3–6]. Projection-based ROMs are based on projecting the governing equations onto a subspace constructed using a limited number of full-order simulations. Thefore, the ROM maintains knowledge of the physical equations while operating in a much lower dimensional subspace than the FOM. This gives ROMs high accuracy and robustness without needing a large training set. Galerkin

*Thermal/Fluid Science and Engineering, MS 9042
†Extreme-Scale Data Science and Analytics, MS 9159, AIAA Member
‡Aerosciences, PO Box 5800, MS 0825, AIAA Member
§Aerosciences, PO Box 5800, MS 0825, AIAA Member

projection [7–9] is the basic projection-based method but can perform poorly for nonlinear systems [10]. Least-squares Petrov-Galerkin projection (LSPG) is a variant that projects the fully discretized equations onto the basis and has been shown to have improved stability and accuracy over Galerkin projection ROMs [5, 10]. Neither Galerkin projection nor LSPG directly enforce conservation laws; both methods may violate conservation laws while minimizing an objective function. Conservative least-squares Petrov-Galerkin (C-LSPG) projection is a modification of LSPG that enforces global conservation over the domain [11, 12]. An additional modification was made by Ref. [12] to develop conservative manifold LSPG (CM-LSPG), which has a nonlinear trial subspace to ensure that physical positivity constraints are satisfied. This work makes use of CM-LSPG ROMs to develop accurate models in hypersonic aerodynamics.

There are few works applying projection-based ROMs to hypersonic applications. In [13–15], the authors apply Galerkin projection ROMs to thermal modeling in hypersonic flows. Blonigan et al. [12] showed that LSPG ROMs are viable in hypersonic aerodynamic applications, demonstrating that ROMs can be over 2000 times faster than the FOM with state errors of approximately 0.1%. However, some ROMs failed to converge due to numerical stability issues.

This work follows on Blonigan et al. [12], in developing robust ROMs for hypersonic aerodynamic applications. While the previous paper (Ref. [12]) demonstrated the basic ROM capability, the goal of this work is to address robustness and efficiency rather than computational speed. We do not apply hyperreduction in this work because hyperreduction improves computational speed but does not improve robustness. If ROMs do not converge without hyperreduction, they will not converge with hyperreduction either, but it is more difficult to isolate the causes of nonconvergence with hyperreduced ROMs. Therefore, to isolate and address the causes of convergence issues hyperreduction is ignored in the preliminary work presented here. The full paper will contain numerical experiments using hyperreduction.

This paper is organized as follows: in § II we describe the governing equations of the full-order models, in § III the reduced order modeling framework of CM-LSPG, in § IV we develop an initial condition subtraction method and various initial guess methods, and describe the initial guess interpolation methods used. In § V we present the numerical results using simulations of the HiFIRE-1 test vehicle. Finally, section VI provides conclusions and possible directions for future work as well as additional contributions that will be presented in the full paper.

II. Full-order model: Governing equations and discretization

This work considers hypersonic aerodynamics in a perfect gas (air), with enthalpy not sufficiently high for dissociation. The governing equations are the perfect gas, compressible Reynolds-Averaged Navier–Stokes (RANS) equations for conserved quantities u_i, where

$$\begin{pmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \\ u_5 \\ u_6 \\ \vdots \\ u_{n_u} \end{pmatrix} = \begin{pmatrix} \rho \\ \rho v_1 \\ \rho v_2 \\ \rho v_3 \\ \rho E \\ \rho \phi_1 \\ \vdots \\ \rho \phi_{n_t} \end{pmatrix} \tag{1}$$

In equation 1, ρ is density of the fluid, ρv_j is the fluid density times the fluid velocity v_j, ρE is the fluid density times the total energy per unit mass E, and ϕ_t is a set of scalars that belongs to turbulent transport equations and n_t is the number of turbulent transport equations. The full set of governing equations for compressible flows is given in [16]. For brevity, they are not shown here.

This work only considers steady-state RANS solutions, which can be expressed in general form as

$$\boldsymbol{f}(\boldsymbol{x}; \boldsymbol{\mu}) = \boldsymbol{0}, \tag{2}$$

where $\boldsymbol{f}$ is the residual vector, $\boldsymbol{x} \in \mathbb{R}^N$ is the state vector and $\boldsymbol{\mu}$ is the vector of system parameters. The steady-state RANS equations are discretized with a cell-centered finite-volume method [17, 18]. Appendix VI.B shows a detailed derivation of the discretized steady-state equations solved.

III. Reduced-order modeling

This section summarizes how projection-based reduced-order models are derived from the governing equations presented in the previous section. Reduced-order models are constructed using a limited number of full-order model solutions in an offline stage and can then be applied at other parameter values during the online stage. During the online stage, ROM evaluations are used for *many-query problems* and should therefore be fast, accurate, and robust. This section details the method of constructing ROMs using conservative manifold least-squares Petrov–Galerkin projection (CM-LSPG).

A. Least-squares Petrov–Galerkin projection

Projection-based reduced-order models estimate at state $\boldsymbol{x}$ with an approximation $\tilde{\boldsymbol{x}} \approx \boldsymbol{x}$ from an affine function

$$\tilde{\boldsymbol{x}}(t;\boldsymbol{\mu}) = \boldsymbol{x}_{\text{ref}}(\boldsymbol{\mu}) + \boldsymbol{\Phi}\hat{\boldsymbol{x}}(t;\boldsymbol{\mu}), \tag{3}$$

where $\boldsymbol{\Phi} \in \mathbb{R}^{N\times p}$ is the reduced-basis matrix of dimension $p \leq N$ and $\hat{\boldsymbol{x}} \in \mathbb{R}^p$ denotes the generalized coordinates. The basis can be computed with a variety of ways, including proper orthogonal decomposition (POD) [19], eigenmode decomposition, or the reduced-basis method [20, 21]. Note that for LSPG, $\boldsymbol{\Phi}$ is not required to be orthonormal, unlike other ROM methods such as Galerkin projection.

For steady-state simulations, LSPG substitutes the approximation $\boldsymbol{x} \leftarrow \tilde{\boldsymbol{x}}$ into the FOM steady-state equations (2), and subsequently minimizes the residual in a weighted ℓ^2-norm, i.e.,

$$\hat{\boldsymbol{x}} = \underset{\hat{\boldsymbol{z}}\in\mathbb{R}^p}{\arg\min} \|\boldsymbol{A}\boldsymbol{f}(\boldsymbol{x}_{\text{ref}}(\boldsymbol{\mu}) + \boldsymbol{\Phi}\hat{\boldsymbol{z}};\boldsymbol{\mu})\|_2. \tag{4}$$

where $\boldsymbol{A} = \boldsymbol{I}$ for ROMs without hyperreduction. With hyperreduction, this weighting matrix $\boldsymbol{A}$ should be sparse in the sense that it has a small number of nonzero columns to ensure that the model incurs an N-independent operation count. See Ref. [6] for details on weighting matrices for hyperreduction.

B. Conservative LSPG projection

Conservative LSPG (C-LSPG) is a modification of LSPG that enforces global conservation as a nonlinear equality constraint [11]. For steady simulations, (Eq. (4)), C-LSPG computes a solution $\hat{\boldsymbol{x}}$ that satisfies

$$\begin{aligned} &\underset{\hat{\boldsymbol{z}}\in\mathbb{R}^p}{\text{minimize}} \ \|\boldsymbol{A}\boldsymbol{f}(\boldsymbol{x}_{\text{ref}}(\boldsymbol{\mu}) + \boldsymbol{\Phi}\hat{\boldsymbol{z}};\boldsymbol{\mu})\|_2 \\ &\text{subject to } \bar{\boldsymbol{C}}\boldsymbol{f}(\tilde{\boldsymbol{x}};\boldsymbol{\mu}) = \boldsymbol{0}. \end{aligned} \tag{5}$$

where $\bar{\boldsymbol{C}} \in \mathbb{R}_+^{n_u\times N}$ is a matrix that maps the residual $\boldsymbol{f}$ to the conservation violations for all n_u conserved quantities. It is derived and defined in appendix VI.C. Note that C-LSPG can also be defined to explicitly enforce conservation on subdomains of the mesh rather than globally. See Ref. [11] for additional details on conservative LSPG projection, including conservation on subdomains, sufficient conditions for feasibility of the associated optimization problems, and *a posteriori* error bounds.

C. Manifold Least-squares Petrov–Galerkin projection

A shortcoming of using a linear trial subspace to compute the approximate state $\tilde{\boldsymbol{x}}$ in (3) is that $\tilde{\boldsymbol{x}}$ can contain non-physical local phenomena for which the solver can not compute residuals, such as negative density or negative temperature. To ensure density, temperature, and turbulence quantities are positive, Ref. [12] introduced manifold LSPG, summarized here.

A nonlinear *trial manifold* $\tilde{\tilde{\boldsymbol{x}}} \approx \boldsymbol{x}$ replaces the linear trial subspace with the form

$$\tilde{\tilde{\boldsymbol{x}}}(t;\boldsymbol{\mu}) = \boldsymbol{h}(\tilde{\boldsymbol{x}}(t;\boldsymbol{\mu})), \tag{6}$$

$\boldsymbol{h}(\tilde{\boldsymbol{x}}) \in \mathbb{R}^N$ is a clipping function, which ensures that density, temperature, and turbulence quantities can only take physical, non-negative values. The clipping function $\boldsymbol{h}(\tilde{\boldsymbol{x}})$ to compute density $\tilde{\tilde{u}}_1$ and turbulence quantities $\tilde{\tilde{u}}_i, i = 1,6,\ldots,n_t$ is

$$\tilde{\tilde{u}}_i = \max(\epsilon_i, \tilde{u}_i), \quad i = 1,6,\ldots,n_t \tag{7}$$

where $\epsilon_i > 0$ is some number that should be very small relative to the free stream $\tilde{\tilde{u}}_i$ and $\tilde{u}_i$ is the corresponding field from $\tilde{\boldsymbol{x}}$. The clipping function for temperature is applied to the conservative variable for energy $\tilde{\tilde{u}}_5$, but the clipping must ensure that temperature rather than energy is positive. The expression relating temperature and energy is

$$c_v T = E - \frac{1}{2}(v_j v_j),$$

where c_v is the constant volume specific heat and v_j is the fluid velocity. Since $c_v > 0$, $T > 0$ can be enforced in $\tilde{\tilde{\boldsymbol{x}}}$ by setting

$$\tilde{\tilde{u}}_5 = \max\left(\epsilon_5 + \frac{1}{2\tilde{\tilde{u}}_1}\left[\tilde{u}_2^2 + \tilde{u}_3^2 + \tilde{u}_4^2\right],\ \tilde{u}_5\right). \tag{8}$$

Note the presence of $\tilde{\tilde{u}}_1$, since the density clipping function must be applied first. Therefore, $\boldsymbol{h}(\tilde{\boldsymbol{x}})$ is of the form $\boldsymbol{h}_5(\boldsymbol{h}_1(\tilde{\boldsymbol{x}}))$, where $\tilde{z} = \boldsymbol{h}_1(\tilde{\boldsymbol{x}})$ applies (7) to $\tilde{\boldsymbol{x}}$, and $\tilde{\tilde{\boldsymbol{x}}} = \boldsymbol{h}_5(\tilde{z})$ applies (8) to $\tilde{z}$.

Steady manifold LSPG modifies the equation 4 by substituting the approximation $\boldsymbol{x} \leftarrow \tilde{\tilde{\boldsymbol{x}}}$ into the FOM steady-state equations (2), and subsequently minimizes the residual in a weighted ℓ^2-norm

$$\hat{\boldsymbol{x}}(\boldsymbol{\mu}) = \underset{\hat{\boldsymbol{z}}\in\mathbb{R}^p}{\arg\min} \left\|\boldsymbol{A}\boldsymbol{f}\left(\boldsymbol{h}(\boldsymbol{x}_{\text{ref}}(\boldsymbol{\mu}) + \boldsymbol{\Phi}\hat{\boldsymbol{z}}(\boldsymbol{\mu})); \boldsymbol{\mu}\right)\right\|_2, \tag{9}$$

The conservation constraint can be applied to the manifold LSPG minimization statement (9), resulting in the conservative manifold LSPG (CM-LSPG) equation

$$\begin{aligned}\hat{\boldsymbol{x}}(\boldsymbol{\mu}) = &\underset{\hat{\boldsymbol{z}}\in\mathbb{R}^p}{\arg\min} \left\|\boldsymbol{A}\boldsymbol{f}\left(\boldsymbol{h}(\boldsymbol{x}_{\text{ref}}(\boldsymbol{\mu}) + \boldsymbol{\Phi}\hat{\boldsymbol{z}}(\boldsymbol{\mu})); \boldsymbol{\mu}\right)\right\|_2 \\ &\text{subject to } \bar{\boldsymbol{C}}\boldsymbol{f}(\tilde{\tilde{\boldsymbol{x}}}; \boldsymbol{\mu}) = \boldsymbol{0}.\end{aligned} \tag{10}$$

IV. Initial guess and reference state methods

The CM-LSPG equation (10) requires both an initial guess $\hat{\boldsymbol{x}}^0(\boldsymbol{\mu})$ and a reference state $\boldsymbol{x}_{\text{ref}}(\boldsymbol{\mu})$ to be defined, and the convergence properties can be strongly dependent on both. This study uses various initial condition guess methods and examines ROM convergence. It is expected that accurate initial guesses will result in faster convergence and more robust ROMs. Performance is evaluated by constructing separate parameter training and test sets. The trial basis $\boldsymbol{\Phi}$ is constructed with full-order models run at the parameters of the training set, but the convergence properties are examined for test set parameters because we are only interested in the predictive scenario.

A. Projected FOM initial guess

An initial guess $\hat{\boldsymbol{x}}^0(\boldsymbol{\mu})$ must be determined for each ROM, but the parameters may be arbitrary. For the training set, an initial guess can be determined by minimizing the norm of the difference between the full-order state $\boldsymbol{x}(\boldsymbol{\mu})$ and the solution in the trial subspace, given by equation 11.

$$\hat{\boldsymbol{x}}^0(\boldsymbol{\mu}) = \underset{\hat{\boldsymbol{z}}\in\mathbb{R}^p}{\arg\min} \|\boldsymbol{x}(\boldsymbol{\mu}) - (\boldsymbol{x}_{\text{ref}}(\boldsymbol{\mu}) + \boldsymbol{\Phi}\hat{\boldsymbol{z}})\|_2. \tag{11}$$

for which the solution is a projection of the full-order state on the reduced basis

$$\hat{\boldsymbol{x}}^0(\boldsymbol{\mu}) = \boldsymbol{\Phi}^+(\boldsymbol{x} - \boldsymbol{x}_{\text{ref}}), \tag{12}$$

where $(\ )^+$ designates a pseudo-inverse. Note that equation (12) holds for a non-orthonormal reduced basis $\boldsymbol{\Phi}$.

When the full-order state $\boldsymbol{x}(\boldsymbol{\mu})$ is unknown, equation 12 cannot be used. Therefore, an interpolation method must be used to identify an initial guess $\hat{\boldsymbol{x}}^0(\boldsymbol{\mu})$. Various approaches have been proposed for applying interpolation between the trial subspaces [3, 4, 22, 23], but those methods only apply to unsteady ROMs. Other approaches have been proposed for interpolation of system matrices of linear time-invariant systems [24, 25], but in this work the system matrices are directly computed within the solver, so those methods do not apply here.

It is possible to interpolate the full-order state $\boldsymbol{x}(\boldsymbol{\mu})$ directly, then used the interpolated state in equation 12 to find the initial guess $\hat{\boldsymbol{x}}^0(\boldsymbol{\mu})$. However, the interpolation must be done in the online stage and since interpolating the full-order

state is time-consuming, that method should be avoided. It is preferable to perform the interpolation in the reduced-state space by first computing the initial guess for each case in the training set, then using those values to interpolate to obtain initial guesses for the test set. The interpolation can be generalized as

$$\hat{\boldsymbol{x}}^0(\boldsymbol{\mu}) = g(\hat{\boldsymbol{x}}^0(\boldsymbol{\mu}_i)_{train}, \boldsymbol{\mu}_i, \boldsymbol{\mu}) \tag{13}$$

where $g()$ is the interpolating function and $\hat{\boldsymbol{x}}^0(\boldsymbol{\mu}_i)_{train}$ are initial guess values at the training set parameters $\boldsymbol{\mu}_i$.

There are two methods of identifying $\hat{\boldsymbol{x}}^0(\boldsymbol{\mu}_i)_{train}$. One method is to directly apply equation 12 on the training set and use the outputs as $\hat{\boldsymbol{x}}^0(\boldsymbol{\mu}_i)_{train}$ for the interpolation. We designate this method as the projected FOM method.

B. Reproductive ROM initial guess

Equation 12 is used to calculate an initial condition for the projected FOM method, but the solutions of the CM-LSPG equation 10 and equation 12 are different. If the basis $\boldsymbol{\Phi}$ is not truncated, the full-state is within the trial subspace; therefore the full state can be exactly reproduced and both equations 12 and 10 will have identical solutions. However, if the basis is truncated, the solutions can be different. In that case, more accurate initial conditions for the test set can be produced by first running ROMs on the training set using initial conditions from equation 12, then using the reduced-state solution of the training set ROMs as $\hat{\boldsymbol{x}}^0(\boldsymbol{\mu}_i)_{train}$ in equation 13 to interpolate initial guesses in the test set. This method is designated the reproductive ROM initial guess method.

C. Interpolation methods

A variety of interpolation methods are available and are tested in this study for interpolating the initial guess (equation 13. A summary of each method is given here. The number of parameter points in the training set is designated as N_p.

1. Inverse Distance Interpolation

Several previous works used inverse distance interpolation to construct the initial guess [12, 26]. Inverse distance interpolation constructs a smooth interpolating function using weights linearly dependent on the inverse distance between each sampling point and training data. The interpolating function is

$$\hat{\boldsymbol{x}}^0(\boldsymbol{\mu}) = \sum_{i=1}^{N_p} \alpha_i \hat{\boldsymbol{x}}^0(\boldsymbol{\mu}_i)_{train} \tag{14}$$

The weights α_i are found with Equation 15

$$\alpha_i = \frac{1/\|\boldsymbol{\mu} - \boldsymbol{\mu}_i\|_2}{\sum_{j=1}^{N_p} 1/\|\boldsymbol{\mu} - \boldsymbol{\mu}_j\|_2} \tag{15}$$

2. Piecewise linear interpolation

Piecewise linear interpolation is constructed by triangulating the known data and applying a linear barycentric interpolation on the vertices of each triangle. The Delaunay triangulation is performed using the Quickhull algorithm presented in Ref. [27], splitting the parameter space into cells. The interpolating surface within each cell is the hyperplane that passes through the known data values at the bounding points. The interpolating function within each cell can be expressed as a linear combination of the function values at the vertices of the cell.

$$\hat{\boldsymbol{x}}^0(\boldsymbol{\mu}) = \sum_{i=1}^{N_{vert}} \alpha_i \hat{\boldsymbol{x}}^0(\boldsymbol{\mu}_i)_{train} \tag{16}$$

where N_{vert} is the number of vertices in each cell, which depends on the dimension of the input. For parameter space with two independent parameters, $N_{vert} = 3$. The weights α_i are found by expressing the coordinate of the sampling point $\boldsymbol{\mu}$ as a linear combination of the training set points $\boldsymbol{\mu}_i$.

$$\boldsymbol{\mu} = \sum_{i=1}^{N_{vert}} \alpha_i \boldsymbol{\mu}_i \tag{17}$$

3. Piecewise Cubic Spline Interpolation

Piecewise cubic spline interpolation finds a smooth, cubic interpolating function. Delaunay triangulation is computed with the Quickhull algorithm [27]. On each triangle, a piecewise cubic polynomial with minimum curvature is computed using a Clough-Tocher scheme [28]. The resulting interpolating function is continuously differentiable over the entire coordinate space. See Refs. [27–29] for details.

It should be noted that for parameter spaces with more than two dimensions, piecewise cubic spline interpolation on scattered data is difficult to implement, and other methods may be preferred.

4. Nearest Neighbor Interpolation

Nearest neighbor interpolation uses the training data point closest in distance to each sampling point to estimate the sampling function. Equation 18 gives the interpolating values

$$\hat{\boldsymbol{x}}^0(\boldsymbol{\mu}) = \hat{\boldsymbol{x}}^0(\boldsymbol{\mu}_s)_{train} \tag{18}$$

where

$$s = \arg\min_i \|\boldsymbol{\mu} - \boldsymbol{\mu}_i\|_2 \tag{19}$$

5. Radial Basis Function interpolation

Radial basis function interpolation uses a weighted sum of radial basis functions to construct a smooth interpolating function, given by Eqn. 20

$$\hat{\boldsymbol{x}}^0_j(\boldsymbol{\mu}) = \sum_{i=1}^{N_p} \alpha_i \Psi(\|\boldsymbol{\mu} - \boldsymbol{\mu}_i\|_2), \tag{20}$$

where $\Psi(b)$ is the radial basis function and $\hat{\boldsymbol{x}}^0_j(\boldsymbol{\mu})$ is the jth component of an initial guess. For this study, a Gaussian basis function is used, where

$$\Psi(b) = e^{-(\varepsilon b)^2}, \tag{21}$$

where ε is the shape parameter.
The weights α_i are found by solving the system

$$K\boldsymbol{\alpha} = \hat{\boldsymbol{x}}^0_j(\boldsymbol{\mu})_{train}, \tag{22}$$

where $\boldsymbol{\alpha} \in \mathbb{R}^{N_p}$ is the vector of weights and

$$K = \begin{bmatrix} \Psi\|\boldsymbol{\mu}_0 - \boldsymbol{\mu}_0\|_2 & .\Psi\|\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0\|_2 & \dots & \Psi\|\boldsymbol{\mu}_{N_p} - \boldsymbol{\mu}_0\|_2 \\ \Psi\|\boldsymbol{\mu}_0 - \boldsymbol{\mu}_1\|_2 & .\Psi\|\boldsymbol{\mu}_1 - \boldsymbol{\mu}_1\|_2 & \dots & \Psi\|\boldsymbol{\mu}_{N_p} - \boldsymbol{\mu}_1\|_2 \\ \vdots & \vdots & \ddots & \vdots \\ \Psi\|\boldsymbol{\mu}_0 - \boldsymbol{\mu}_{N_p}\|_2 & \Psi\|\boldsymbol{\mu}_1 - \boldsymbol{\mu}_{N_p}\|_2 & \dots & \Psi\|\boldsymbol{\mu}_{N_p} - \boldsymbol{\mu}_{N_p}\|_2 \end{bmatrix} \tag{23}$$

The shape parameter ε in Eqn. 21 controls the width of the Gaussian basis function. For $\varepsilon \to 0$, the basis functions are very narrow and the interpolation function is zero everywhere except close to the known points x_i. For $\varepsilon \to \infty$, the matrix in Eqn. 23 is ill-conditioned. In practice, the shape parameter should be set as high as possible while maintaining a condition number sufficiently below numerical precision [30]. In this work, the shape parameter is initially set to 10^{-2} and iteratively doubled until the condition number is above 10^{12}. Computing the weights α_i can be done in the offline stage, keeping the online computational costs low.

6. Polyharmonic spline interpolation

Polyharmonic spline interpolation is a linear combination of radial basis functions and a polynomial term [31, 32]. The polynomial term improves fitting accuracy away from the interpolating points $\boldsymbol{\mu}_i$ in comparison to radial basis function interpolation. The interpolating function is

$$\hat{\boldsymbol{x}}_j^0(\boldsymbol{\mu}) = \sum_{i=1}^{N_p} \alpha_i \Psi(\|\boldsymbol{\mu} - \boldsymbol{\mu}_i\|_2) + v^T \begin{bmatrix} 1 \\ \boldsymbol{\mu} \end{bmatrix} \tag{24}$$

where $v \in \mathbb{R}^{d_{\boldsymbol{\mu}}}$ is a vector of the polynomial weights, where $d_{\boldsymbol{\mu}}$ is the dimension of the parameter space $\boldsymbol{\mu}$. $\Psi(b)$ is a radial basis function of the form

$$\Psi(b) = \begin{cases} b^k & \text{with } k = 1, 3, 5, \ldots \\ b^k ln(b) & \text{with } k = 2, 4, 6, \ldots \end{cases} \tag{25}$$

In this work, $k = 2$ whenever polyharmonic interpolation is used. The weights are found by solving the system

$$\begin{bmatrix} K & L \\ L^T & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{\alpha} \\ v \end{bmatrix} = \begin{bmatrix} \hat{\boldsymbol{x}}_j^0(\boldsymbol{\mu})_{train} \\ 0 \end{bmatrix} \tag{26}$$

where K is defined in equation 23 and L is

$$L = \begin{bmatrix} 1 & 1 & \ldots & 1 \\ \boldsymbol{\mu}_0 & \boldsymbol{\mu}_1 & \ldots & \boldsymbol{\mu}_{N_p} \end{bmatrix}^T \tag{27}$$

7. Preprocessing for interpolation

This study considers a case with two independent parameters. For a multi-dimensional parameter space, each parameter must be normalized before applying interpolation. Without normalization, interpolation methods that use distance between points to calculate weighting can result in widely differing sensitivity to the different parameters. For example, if the variance of one parameter is much lower than the others, inverse distance interpolation would be much more sensitive to that parameter. To prevent this, each parameter is mean subtracted and divided by the standard deviation before applying interpolation.

V. Results (preliminary)

The numerical experiments presented here use two codes developed at Sandia National Laboratories. SPARC (Sandia Parallel Aerodynamics and Reentry Code) [33] is a compressible CFD code for aerodynamics and aerothermodynamics, and has capabilities to solve the compressible Navier-Stokes and Reynolds-Averaged-Navier-Stokes (RANS) equations on structured or unstructured grids with a finite volume discretization scheme. SPARC also has capabilities for solving the heat equation, physical chemistry, and associated equations for non-decomposing and decomposing ablators.

The ROMs are run using *Pressio*[*], an open-source C++11 library to enable parallel, scalable ROM capabilities for any C++ simulation code [34]. *Pressio* interfaces with a solver with a minimally intrusive API that requires only the residual $\boldsymbol{f}(\boldsymbol{x}, t; \boldsymbol{\mu})$ and Jacobian matrix $\partial \boldsymbol{f}(\boldsymbol{x}, t; \boldsymbol{\mu})/\partial \boldsymbol{x}$ for a given state $\boldsymbol{x}$, time t, and parameters $\boldsymbol{\mu}$. *Pressio* and SPARC have previously been interfaced and used to run CM-LSPG ROMs in Ref. [12], with demonstrated speedup hundreds to thousands of times faster than full-order models.

This work studies ROMs on a hypersonic flow configuration using CM-LSPG and examines the convergence properties when the initial condition method, interpolation method, and freestream subtraction are varied. The preliminary results shown below do not use hyperreduction, but the full paper will include numerical experiments using hyperreduction.

A. Flow configuration

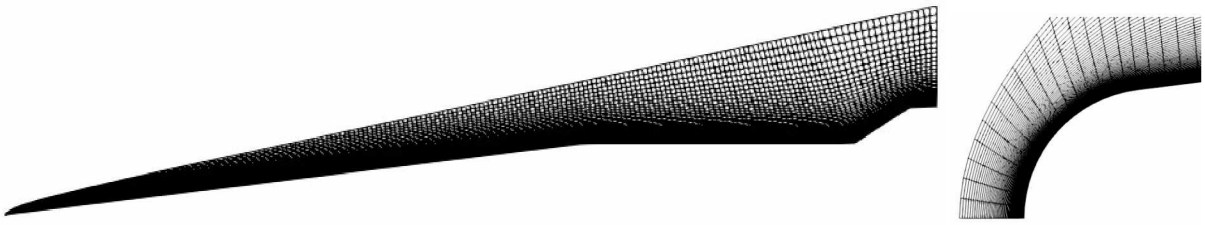
A 2-dimensional hypersonic configuration is used to test the methods presented here. The configuration is the HiFIRE-1 (Hypersonic International Flight Research Experimentation) vehicle [35]. The baseline case is run 30 of the experimental results presented in Ref. [36]. The freestream conditions are presented in table 1. Because the geometry is axisymmetric with a zero angle-of-attack, the full-order models require a mesh only on one side of the centerline, shown in figure 1. The mesh has 32,768 cells, corresponding to a state-space size of 196,608 since there are 6 conserved quantities ($n_u = 6$). Figure 2 shows a slice of the flow colored by density.

[*]https://github.com/Pressio

Density	0.04 - 0.06 kg/m^3
Velocity	1810.0 - 2413.4 m/s
Mach Number	6 - 8
Angle of attack	0.0°
Temperature	226.46 K
Reynolds Number (baseline)	1×10^7 $1/m$

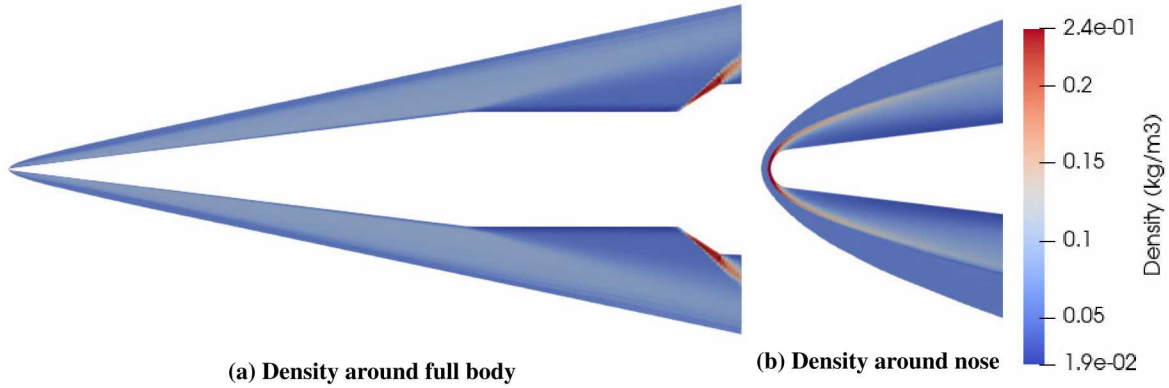
Table 1 Free stream flow conditions for the HiFIRE-1 vehicle results presented here.

The full-order models are run with the Spalart-Allmaras turbulence model [37]. The flow solver uses pseudo-time stepping with a backward Euler time step and scheduled increases in CFL number. The convergence criteria are a reduction in relative residual by 5 orders of magnitude or 25,000 iterations.



(a) Full mesh **(b) Mesh around nose**

Fig. 1 HiFIRE-1 Mesh.



(a) Density around full body **(b) Density around nose**

Fig. 2 SPARC simulation of HIFiRE-1 at $\rho = 0.0568$, $Ma = 6.12$. The flow field is colored by density ρ.

B. ROM setup

The ROM parameter space is defined by variations in Mach number and density given in table 1. The training set has a total of 12 different points in parameter space, and the test set has 30 points. The training set points in parameter space are chosen by Latin hypercube sampling for 8 of the points, and the four corners of parameter space make up the remaining 4 points. The test set is determined with Latin hypercube sampling for all 30 points. Figure 3 shows the parameters of both sets. The parameter space has wider density and Mach number ranges in comparison to Ref. [12], and the training set is smaller.

The ROM basis is The CM-LSPG equation (equation 10) is solved with the normal equations within *Pressio*. The solver is run until the relative residual L2 norm is below 10^{-4} or after 100 iterations. The clipping functions are set at $\epsilon_1 = \epsilon_5 = \epsilon_6 = 10^{-8}$.

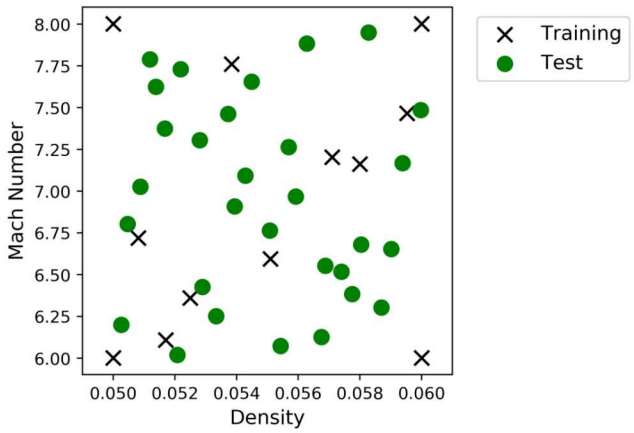


Fig. 3 Test and trainings set locations in parameter space

1. Error metrics

The ROMs are analyzed using a number or error metrics. The state error is defined as

$$\mathcal{E}_{\boldsymbol{x}} = \frac{\|\boldsymbol{x}(\boldsymbol{\mu}) - \tilde{\boldsymbol{x}}(\boldsymbol{\mu})\|_2}{\|\boldsymbol{x}(\boldsymbol{\mu})\|_2} \tag{28}$$

where $\boldsymbol{x}(\boldsymbol{\mu})$ is the full state computed with the FOM and $\tilde{\boldsymbol{x}}(\boldsymbol{\mu})$ is an approximated state. The vector $\tilde{\boldsymbol{x}}(\boldsymbol{\mu})$ is usually the ROM solution, but we also compute $\tilde{\boldsymbol{x}}(\boldsymbol{\mu})$ for other approximated states as well. One other approximated state is the FOM solution projected on the basis $\boldsymbol{\Phi}$, given by

$$\tilde{\boldsymbol{x}}_{FOM}(\boldsymbol{\mu}) \equiv \boldsymbol{\Phi}(\boldsymbol{\Phi})^{+}(\boldsymbol{x}(\boldsymbol{\mu}) - \boldsymbol{x}_{\text{ref}}) + \boldsymbol{x}_{\text{ref}}. \tag{29}$$

$\tilde{\boldsymbol{x}}_{FOM}(\boldsymbol{\mu})$ is a lower bound for the ROM state error, since it is the most accurate approximation of the FOM state contained in the basis $\boldsymbol{\Phi}$. The state error (equation 28) is also computed for the initial guess state as well.

Two output quantities that are computed from a state are the integrated wall heat flux $Q_{wall}(\boldsymbol{\mu})$ and streamwise forces $\boldsymbol{F}_{\boldsymbol{x}}(\boldsymbol{\mu})$ The wall heat flux error is defined as

$$\varepsilon_{Q_{wall}} = \frac{|Q_{wall}(\boldsymbol{\mu}) - \tilde{Q}_{wall}(\boldsymbol{\mu})|}{|Q_{wall}(\boldsymbol{\mu})|} \tag{30}$$

The force error is defined as

$$\varepsilon_{F_x} = \frac{|\boldsymbol{F}_x(\boldsymbol{\mu}) - \tilde{\boldsymbol{F}}_x(\boldsymbol{\mu})|}{|\boldsymbol{F}_x(\boldsymbol{\mu})|} \tag{31}$$

In addition to the errors defined in equations 28 - 31, convergence is quantified by the number of failed cases, number of unconverged cases, and mean iterations. The ROM solver can fail to produce a solution if values exceed numerical bounds. Alternatively, the ROMs can produce a solution but remain unconverged. Here convergence is defined by the relative L2 residual decreasing below 10^{-4} within 100 iterations. Finally, the mean iterations is the mean number of iterations of cases that did not fail. Note that if some cases in the set remain unconverged, the mean iterations will be affected by the maximum number of iterations (set here as 100 iterations).

We consider ROMs with freestream subtraction, reproductive ROM initial condition, and polyharmonic interpolation as a baseline ROM setup. This baseline setup contains all the improvements presented in this work. Presenting the results of a full sweep over all different setups would be excessive, so for clarity we show a limited number of different setups.

C. Effect of number of basis vectors

The number of basis vectors p is an important parameter of ROMs dependent on the specific problem. The number of basis vectors used should be large enough that most of the variance of the training set is captured when the training set is projected on the trial basis. There are 12 FOM's in the training set, but only 11 basis vectors are needed to span the

Table 2 Convergence properties and error on the test set with various number of modes. All cases have a reproductive ROM initial guess and polyharmonic interpolation. Errors are defined as percentages, where $\%\epsilon = 100\epsilon$. Errors are labeled as IG for the initial guess, ROM for CM-LSPG, and FOM for the error obtained by projecting the FOM solution $x(\mu)$ on the basis Φ as in (29). Error entries are colored by their error value: red for $\varepsilon_x > 4\%$, orange for $2\% < \varepsilon_x \leq 4\%$, yellow for $1\% < \varepsilon_x \leq 2\%$, and green for $\varepsilon_x \leq 1\%$.

Number of modes p	# failed cases	# unconverged cases	Mean iterations	$\%\mathcal{E}_x$			$\%\varepsilon_{Q_{wall}}$		$\%\varepsilon_{F_x}$	
				IG	ROM	FOM	IG	ROM	IG	ROM
3	0	0	21.5	1.04	0.95	0.90	3.84	3.88	0.15	0.21
5	0	0	22.9	0.90	0.66	0.61	2.81	4.40	0.23	0.20
8	0	2	31.3	0.90	0.60	0.47	1.17	2.88	0.19	0.14
11	0	0	27.4	0.92	0.57	0.37	1.19	1.77	0.20	0.12

training set subspace since the data has zero mean after the reference vector is subtracted. Figure 4 shows the fraction of variance captured versus the number of basis modes used. We test ROMs separately using a trial basis truncated to $p = 3$, 5, 8, and 11 basis vectors, for which the variance captured is 99.88%, 99.97%, 99.99%, and 100%, respectively.

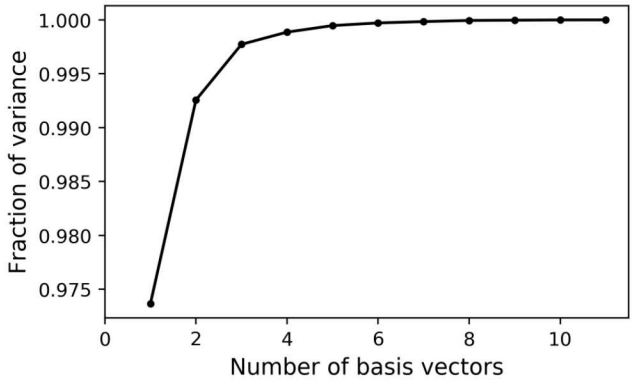


Fig. 4 Fraction of variance contained by basis versus number of basis vectors

The effect of number of modes on the baseline setup (ROM reconstruction and polyharmonic interpolation) is evaluated with the error metrics in section V.B.1.

Table 2 shows averaged statistics over the 30 test set ROMs for each number of basis vectors. None of the cases failed; however 2 of the 30 cases did not converge for $p = 8$. The number of iterations varies between 21 and 32 iterations for the different setups. Because of the unconverged cases, the mean iterations is larger for $p = 8$ than the other setups.

The initial guess state errors do not vary significantly with p, and are all between 0.90% and 1.04%. For all p, the ROMs decrease the state error from the initial guess to values close to the FOM state error. The FOM errors decrease with increasing p as a larger basis contains more variance in the test set, but the FOM errors are not zero because the trial basis was calculated from training set data.

The ROM mean integrated heat flux error $\varepsilon_{Q_{wall}}$ is between 1% and 5% for each p, but does not exhibit a consistent trend with the number of basis vectors as $\varepsilon_{Q_{wall}}$ is largest for $p = 5$ and smallest for $p = 11$. For all p, the ROMs increase $\varepsilon_{Q_{wall}}$ from the initial guess. The mean streamwise force error ε_{F_x} is very low (below 0.21%) for all setups and generally decreases with increasing p. For the higher numbers of basis vectors p, the ROM streamwise force error ε_{F_x} is improved over the initial guess.

D. Effect of interpolation method

Table 3 shows statistics of the test set ROMs with different interpolation methods. All the interpolation methods except polyharmonic interpolation have unconverged cases. All three error metrics of the initial guess are highest for inverse distance interpolation and second highest for nearest neighbor interpolation. The error metrics of the ROMs are

Table 3 Convergence properties and error on the test set with the interpolation methods described in section IV.C. All cases have a reproductive ROM initial guess and $p = 11$ basis vectors. Errors are defined as percentages, where $\%\epsilon = 100\epsilon$. Errors are labeled as IG for the initial guess, ROM for CM-LSPG, and FOM for the error obtained by projecting the FOM solution $x(\mu)$ on the basis Φ as in (29). Error entries are colored by their error value: red for $\varepsilon_x > 4\%$, orange for $2\% < \varepsilon_x \leq 4\%$, yellow for $1\% < \varepsilon_x \leq 2\%$, and green for $\varepsilon_x \leq 1\%$.

Interpolation method	# Cases failed	# Cases unconverged	Mean iterations	$\%\mathcal{E}_x$			$\%\varepsilon_{Q_{wall}}$		$\%\varepsilon_{F_x}$	
				IG	ROM	FOM	IG	ROM	IG	ROM
Inverse dist.	0	2	32.6	10.82	0.57	0.37	14.51	1.73	8.72	0.11
Linear	0	2	32.2	1.09	0.57	0.37	2.64	1.77	0.34	0.12
Cubic	0	2	32.5	0.88	0.57	0.37	0.87	1.76	0.13	0.12
Nearest N.	0	3	37.4	8.01	0.57	0.37	9.50	1.72	6.47	0.11
RBF	0	1	26.1	2.36	0.57	0.37	0.16	1.75	0.17	0.12
Polyharm.	0	0	27.4	0.92	0.57	0.37	1.19	1.77	0.20	0.12

Table 4 Convergence properties and error on the test set with reproductive ROM and projected FOM initial guess methods for various number of modes p. All cases have polyharmonic interpolation. The initial guess methods are labeled RR for reproductive ROM and PF for projected FOM. Errors are defined as percentages, where $\%\epsilon = 100\epsilon$. Errors are labeled as IG for the initial guess, ROM for CM-LSPG, and FOM for the error obtained by projecting the FOM solution $x(\mu)$ on the basis Φ as in (29). Error entries are colored by their error value: red for $\varepsilon_x > 4\%$, orange for $2\% < \varepsilon_x \leq 4\%$, yellow for $1\% < \varepsilon_x \leq 2\%$, and green for $\varepsilon_x \leq 1\%$.

IG method	p	# Cases failed	# Cases unconverged	Mean iterations	$\%\mathcal{E}_x$			$\%\varepsilon_{Q_{wall}}$		$\%\varepsilon_{F_x}$	
					IG	ROM	FOM	IG	ROM	IG	ROM
RR	**5**	0	0	22.9	0.90	0.66	0.61	2.81	4.40	0.23	0.20
PF	**5**	0	0	25.2	0.91	0.66	0.61	2.56	4.40	0.20	0.20
RR	**11**	0	0	27.4	0.92	0.57	0.37	1.19	1.77	0.20	0.12
PF	**11**	0	2	32.8	0.92	0.57	0.37	1.20	1.82	0.20	0.12

almost the same for all interpolation methods, and it was verified that the ROMs solutions are identical to 2 decimal places for the cases that converged whichever interpolation method is used. The radial basis function interpolation and polyharmonic interpolation have the lowest mean iterations.

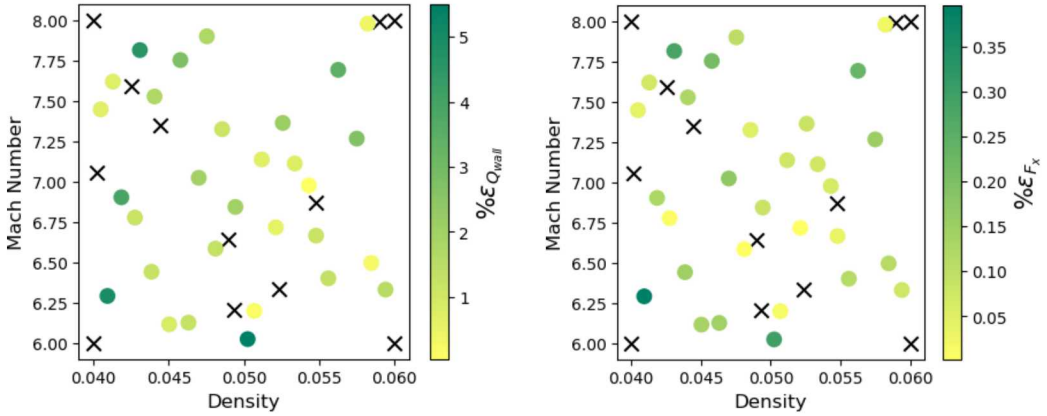
E. Effect of initial guess method

Table 4 shows tabulated results of the two initial guess methods, with comparisons for various $p = 5$ and $p = 11$. For $p = 5$, the ROM state error $\mathcal{E}_x$, heat flux error $\varepsilon_{Q_{wall}}$, and force error ε_{F_x} are identical, so the end states are the same whichever initial guess method is used. However, for $p = 11$ with the projected FOM method, two of the cases did not converge. The reconstructive ROM initial guess method has slightly fewer mean iterations than the projected FOM method. Note that the initial guess state error $\mathcal{E}_x$ is larger for the reconstructive ROM method than the projected FOM method because the projected FOM method first optimizes the state error in the training set before interpolation, as described in section IV.A.

Figure 5 shows the integrated heat flux error and streamwise force errors for each ROM in parameter space for the setup with $p = 11$, polyharmonic interpolation, and reconstructive ROM initial guess. The ROMs near the center of the parameter space are generally more accurate than ROMs near the edges of the parameter set. This suggests that it is unlikely that ROMs that extrapolate (contain parameters outside the range of the training set) would be able to make accurate predictions. Also note that the ROMs which have a large integrated heat flux error are generally the same ones that have a large streamwise force error. This is likely because both output quantities will be inaccurate if the ROM state near the vehicle body is inaccurate.

The mean heat flux errors $\varepsilon_{Q_{wall}}$ show that for this setup, on average it is better to use an accurate interpolation

method to identify heat flux than to use ROMs. This may not be the case for all problems, and the full paper will examine more cases.



(a) Integrated heat flux error $\%\varepsilon_{Q_{wall}}$ **(b) Streamwise force error $\%\varepsilon_{F_x}$**

Fig. 5 Integrated heat flux and streamwise force error for ROMs with $p = 11$, polyharmonic interpolation, and reconstructive ROM initial guess. The $\times$ marker denotes a training case and the colored circled denote test cases.

VI. Conclusions

Robustness can be an issue for ROMs in hypersonic flows, and this work develops improvements to the initial guess calculation that enhance robustness in hypersonic flight. Conservative manifold least-squares Petrov-Galerkin (CM-LSPG) is applied to the HiFIRE-1 vehicle in hypersonic flight with varying Mach number and density.

It is important to have an accurate initial guess for the ROMs to converge. This work presents numerical experiments with different initial guess methods. The reconstructive ROM method developed here improves the initial guess and speeds up convergence. This work showed that the interpolation method can make a major difference to errors of the initial guess. In particular, the inverse distance interpolation used for ROMs in Refs. [12, 26] is the least accurate interpolation method. The polyharmonic and radial basis function interpolations result in significantly faster convergence. It is also important to note that with inverse distance interpolation, the ROMs significantly increase accuracy of integrated heat flux in comparison to the initial guess, but with more accurate interpolation methods the ROMs actually decrease integrated heat flux accuracy. The ROMs increase accuracy of streamwise force for each interpolation method, but by differing amounts. Both quantities are more accurate for ROMs near the center of the parameter space than near the edges. It is important for future works to compare output quantities of ROMs to the output quantities with an accurate interpolation method.

It is important to recognize that even with these improvements, ROMs can still have robustness issues. The ROMs presented here have a relatively narrow density range, and significantly expanding the density range without increasing the number of training set simulations can result in ROMs that fail or do not converge. Future work is needed to continue to address robustness issues. In addition to the studies presented here, the full paper will contain:

- Development of a freestream subtraction method to increase ROM accuracy
- ROMs on a larger three-dimensional mesh
- Tests of CM-LSPG with hyperreduction

There are a number of possible directions future works could pursue. Future work could be directed at developing preconditioners for CM-LSPG solvers. Future work could also examine alternative nonlinear mappings to replace POD modes for hypersonics, as a linear basis has limitations in advection-dominated flows. The integrated heat flux errors in the ROMs presented here are significantly larger than state errors or streamwise force errors, and future works could target improving calculations of surface quantities to improve heat flux predictions. With continued research, ROMs have potential as robust, accurate, and fast alternatives to full-order models for hypersonic vehicles.

Appendix

A. Physical conservation laws

This work considers parameterized systems of physical conservation laws. The governing equations in integral form can be expressed as

$$\frac{d}{dt}\int_{\omega} u_i(\vec{x},t;\boldsymbol{\mu})\,\mathrm{d}\vec{x} + \int_{\gamma} \boldsymbol{g}_i(\vec{x},t;\boldsymbol{\mu})\cdot\boldsymbol{n}(\vec{x})\,\mathrm{d}\vec{s}(\vec{x}) = \int_{\omega} s_i(\vec{x},t;\boldsymbol{\mu})\,\mathrm{d}\vec{x}, \quad i\in\mathbb{N}(n_u),\ \forall\omega\subseteq\Omega, \tag{32}$$

$u_i(\vec{x},t;\boldsymbol{\mu})$ are the conserved quantities, which are functions of location $\vec{x}$, time t, and parameters $\boldsymbol{\mu}$. Equation 32 is solved over the time domain given an initial condition denoted by $u_i^0\in\mathbb{R}$ such that $u_i(\vec{x},0;\boldsymbol{\mu}) = u_i^0(\vec{x};\boldsymbol{\mu})$, $i\in\mathbb{N}(n_u)$, where $\mathbb{N}(a) := \{1,\dots,a\}$. $\boldsymbol{g}_i$ is the fluxes associated with the ith conserved variable, and s_i is the source term of the ith conserved variable. Here, ω denotes any subset of the spatial domain of interest $\Omega\subset\mathbb{R}^d$ with $d\le 3$; $\gamma := \partial\omega$ denotes the boundary of the subset ω, while $\Gamma := \partial\Omega$ denotes the boundary of the domain Ω; $\mathrm{d}\vec{s}(\vec{x})$ denotes integration with respect to the boundary. $\boldsymbol{n}:\gamma\to\mathbb{R}^d$ denotes the outward unit normal to γ. We emphasize that any conservation law can be expressed in the form of equations (32).

B. Finite-volume discretization

The governing equations (32) are discretized with the finite-volume method [17, 18], as it explicitly enforces conservation over prescribed control volumes. A mesh $\mathcal{M}$ partitions the spatial domain $\Omega\subset\mathbb{R}^d$ with $d\le 3$ into $N_\Omega\in\mathbb{N}$ non-overlapping (closed, connected) control volumes $\Omega_i\subseteq\Omega$, $i\in\mathbb{N}(N_\Omega)$. We define the mesh as $\mathcal{M} := \{\Omega_i\}_{i=1}^{N_\Omega}$, and denote the boundary of the ith control volume by $\Gamma_i := \partial\Omega_i$. Each control-volume boundary Γ_i is further partitioned into a set of faces. Enforcing conservation (32) on each control volume in the mesh yields

$$\frac{d}{dt}\int_{\Omega_j} u_i(\vec{x},t;\boldsymbol{\mu})\,\mathrm{d}\vec{x} + \int_{\Gamma_j} \boldsymbol{g}_i(\vec{x},t;\boldsymbol{\mu})\cdot\boldsymbol{n}_j(\vec{x})\,\mathrm{d}\vec{s}(\vec{x}) = \int_{\Omega_j} s_i(\vec{x},t;\boldsymbol{\mu})\,\mathrm{d}\vec{x}, \quad i\in\mathbb{N}(n_u),\ j\in\mathbb{N}(N_\Omega), \tag{33}$$

where $\boldsymbol{n}_j\in\mathbb{R}^d$ denotes the unit normal to control surface Γ_j. The finite-volume scheme forms a state vector of the mean conserved quantities in each cell $\boldsymbol{x}\in\mathbb{R}^N$ with $N = N_\Omega n_u$ such that

$$x_{\mathcal{I}(i,j)}(t;\boldsymbol{\mu}) = \frac{1}{|\Omega_j|}\int_{\Omega_j} u_i(\vec{x},t;\boldsymbol{\mu})\,\mathrm{d}\vec{x}, \quad i\in\mathbb{N}(n_u),\ j\in\mathbb{N}(N_\Omega), \tag{34}$$

where $\mathcal{I}:\mathbb{N}(n_u)\times\mathbb{N}(N_\Omega)\to\mathbb{N}(N)$ denotes a mapping from conservation-law index and control-volume index to degree of freedom. The finite-volume scheme also has a velocity vector $\boldsymbol{f}(\boldsymbol{x},t;\boldsymbol{\mu}) = \boldsymbol{f}^g(\boldsymbol{x},t;\boldsymbol{\mu}) + \boldsymbol{f}^s(\boldsymbol{x},t;\boldsymbol{\mu})$ with $\boldsymbol{f}^g,\boldsymbol{f}^s\in\mathbb{R}^N$ whose elements consist of

$$\begin{aligned} f^g_{\mathcal{I}(i,j)}(\boldsymbol{x},t;\boldsymbol{\mu}) &= -\frac{1}{|\Omega_j|}\int_{\Gamma_j} \boldsymbol{g}_i^{\mathrm{FV}}(\boldsymbol{x};\vec{x},t;\boldsymbol{\mu})\cdot\boldsymbol{n}_j(\vec{x})\,\mathrm{d}\vec{s}(\vec{x}),\\ f^s_{\mathcal{I}(i,j)}(\boldsymbol{x},t;\boldsymbol{\mu}) &= \frac{1}{|\Omega_j|}\int_{\Omega_j} s_i^{\mathrm{FV}}(\boldsymbol{x};\vec{x},t;\boldsymbol{\mu})\,\mathrm{d}\vec{x}, \end{aligned}$$

for $i\in\mathbb{N}(n_u)$, $j\in\mathbb{N}(N_\Omega)$. Here, the fields $\boldsymbol{g}_i^{\mathrm{FV}}\in\mathbb{R}^d$ and $s_i^{\mathrm{FV}}\in\mathbb{R}$, $i\in\mathbb{N}(n_u)$ denote the approximated flux and source, respectively, associated with the ith conserved variable (per unit area per unit time). Substituting the finite-volume state vector $\int_{\Omega_j} u_i(\vec{x},t;\boldsymbol{\mu})\,\mathrm{d}\vec{x} \leftarrow |\Omega_j| x_{\mathcal{I}(i,j)}(t;\boldsymbol{\mu})$ and associated fluxes and sources $\boldsymbol{g}_i \leftarrow \boldsymbol{g}_i^{\mathrm{FV}}$, and $s_i \leftarrow s_i^{\mathrm{FV}}$ in Eq. (33) and dividing by $|\Omega_j|$ yields

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x},t;\boldsymbol{\mu}), \qquad \boldsymbol{x}(0;\boldsymbol{\mu}) = \boldsymbol{x}^0(\boldsymbol{\mu}), \tag{35}$$

where $x^0_{\mathcal{I}(i,j)}(\boldsymbol{\mu}) := \frac{1}{|\Omega_i|}\int_{\Omega_j} u_i^0(\vec{x};\boldsymbol{\mu})\,\mathrm{d}\vec{x}$ denotes the parameterized initial condition. This is a parameterized system of nonlinear ordinary differential equations (ODEs) defining an initial value problem, which we consider to be our full-order model (FOM). We hereafter refer to Eq. (35) as the FOM ODE.

For steady-state problems, there is no time dependence so the FOM ODE (35) reduces to

$$\boldsymbol{f}(\boldsymbol{x};\boldsymbol{\mu}) = \boldsymbol{0},$$

which we refer to as the FOM steady-state equations. Here, we have abused notation and set $\boldsymbol{f}(\boldsymbol{x};\boldsymbol{\mu}) = \boldsymbol{f}(\boldsymbol{x},t;\boldsymbol{\mu})$. For steady-state problems, we also refer to $\boldsymbol{f}(\boldsymbol{x};\boldsymbol{\mu})$ as the residual.

C. Conservation constraints

The conservation constraint directly enforces global conservation on the mesh, and has been shown to significantly improve accuracy over standard LSPG [11, 12]. While the conservation constraint can be applied to subdomains of the mesh [11], there is no obvious way to partition the mesh so this work uses a global conservation constraint.

The conservation law (32) enforced globally is

$$\frac{d}{dt}\int_{\Omega} u_i(\vec{x},t;\boldsymbol{\mu})\,\mathrm{d}\vec{x} + \int_{\Gamma} \boldsymbol{g}_i(\vec{x},t;\boldsymbol{\mu})\cdot\bar{\boldsymbol{n}}(\vec{x})\,\mathrm{d}\vec{s}(\vec{x}) = \int_{\Omega} s_i(\vec{x},t;\boldsymbol{\mu})\,\mathrm{d}\vec{x}, \quad i\in\mathbb{N}(n_u) \tag{36}$$

where $\bar{\boldsymbol{n}}\in\mathbb{R}^d$ is the outward unit normal vector of the global control surface Γ. To derive the conservation constraint, the same finite-volume discretization employed earlier is applied to the global mesh. A global state vector $\bar{\boldsymbol{x}}\in\mathbb{R}^{n_u}$ is defined as

$$\bar{\boldsymbol{x}}_i(t;\boldsymbol{\mu}) = \frac{1}{\Omega}\int_{\Omega} u_i(\vec{x},t;\boldsymbol{\mu})\,\mathrm{d}\vec{x}, \quad i\in\mathbb{N}(n_u) \tag{37}$$

where $\bar{\boldsymbol{x}}_i$ is the ith component of the global state vector $\bar{\boldsymbol{x}}$. The global state vector $\bar{\boldsymbol{x}}$ can be computed from the state vector $\boldsymbol{x}$ as

$$\bar{\boldsymbol{x}}(\boldsymbol{x}) = \bar{\boldsymbol{C}}\boldsymbol{x},$$

where $\bar{\boldsymbol{C}}\in\mathbb{R}_{+}^{n_u\times N}$ has elements $\bar{\boldsymbol{C}}_{i,\mathcal{I}_{(k,j)}} = \frac{|\Omega_j|}{|\Omega|}\delta_{ik}$.

Similarly, the residual associated with the finite-volume scheme applied to global conservation can be expressed as

$$\bar{\boldsymbol{f}}(\boldsymbol{x};\boldsymbol{\mu}) = \bar{\boldsymbol{C}}\boldsymbol{f}(\boldsymbol{x};\boldsymbol{\mu}), \tag{38}$$

such that subdomain conservation can be expressed as

$$\bar{\boldsymbol{C}}\boldsymbol{f}(\boldsymbol{x};\boldsymbol{\mu}) = 0. \tag{39}$$

For the detailed explanation on the derivation of Eqs. (38)–(39), we refer readers Ref. [11, Section 4.1].

Acknowledgments

This paper describes objective technical results and analysis. Any subjective views or opinions that might be expressed in the paper do not necessarily represent the views of the U.S. Department of Energy or the United States Government. Supported by the Laboratory Directed Research and Development program at Sandia National Laboratories, a multimission laboratory managed and operated by National Technology and Engineering Solutions of Sandia, LLC., a wholly owned subsidiary of Honeywell International, Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA-0003525.

References

[1] Slotnick, J., Khodadoust, A., Alonso, J., Darmofal, D., Gropp, W., Lurie, E., and Mavriplis, D., "CFD vision 2030 study: a path to revolutionary computational aerosciences," 2014.

[2] Yondo, R., Andrés, E., and Valero, E., "A review on design of experiments and surrogate models in aircraft real-time and many-query aerodynamic analyses," *Progress in Aerospace Sciences*, Vol. 96, 2018, pp. 23–61.

[3] Lieu, T., Farhat, C., and Lesoinne, M., "POD-based aeroelastic analysis of a complete F-16 configuration: ROM adaptation and demonstration," *46th AIAA/ASME/ASCE/AHS/ASC structures, structural dynamics and materials conference*, 2005, p. 2295.

[4] Amsallem, D., and Farhat, C., "Interpolation method for adapting reduced-order models and application to aeroelasticity," *AIAA journal*, Vol. 46, No. 7, 2008, pp. 1803–1813.

[5] Carlberg, K. T., Bou-Mosleh, C., and Farhat, C., "Efficient non-linear model reduction via a least-squares Petrov–Galerkin projection and compressive tensor approximations," *International Journal for Numerical Methods in Engineering*, Vol. 86, No. 2, 2011, pp. 155–181.

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

[7] Rowley, C. W., Colonius, T., and Murray, R. M., "Model reduction for compressible flows using POD and Galerkin projection," *Physica D Nonlinear Phenomena*, Vol. 189, No. 1-2, 2004, pp. 115–129.

[8] Terragni, F., Valero, E., and Vega, J. M., "Local POD plus Galerkin projection in the unsteady lid-driven cavity problem," *SIAM Journal on Scientific Computing*, Vol. 33, No. 6, 2011, pp. 3538–3561.

[9] Barone, M. F., Kalashnikova, I., Segalman, D. J., and Thornquist, H. K., "Stable Galerkin reduced order models for linearized compressible flow," *Journal of Computational Physics*, Vol. 228, No. 6, 2009, pp. 1932–1946.

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

[11] Carlberg, K. T., Choi, Y., and Sargsyan, S., "Conservative model reduction for finite-volume models," *Journal of Computational Physics*, Vol. 371, 2018, p. 280–314.

[12] Blonigan, P. J., Carlberg, K., Rizzi, F., Howard, M., and Fike, J. A., "Model reduction for hypersonic aerodynamics via conservative LSPG projection and hyper-reduction," *AIAA Scitech 2020 Forum*, 2020, p. 0104.

[13] Falkiewicz, N. J., and Cesnik, C. E. S., "Proper Orthogonal Decomposition for Reduced-Order Thermal Solution in Hypersonic Aerothermoelastic Simulations," *AIAA Journal*, Vol. 49, No. 5, 2011, pp. 994, 1010. doi:10.2514/1.J050701.

[14] Klock, R. J., and Cesnik, C. E. S., "Nonlinear Thermal Reduced-Order Modeling for Hypersonic Vehicles," *AIAA Journal*, Vol. 55, No. 7, 2017, pp. 2358, 2369. doi:10.2514/1.J055499.

[15] Mignolet, M. P., Culler, A. J., McNamara, J. J., Matney, A., and Spottswood, S. M., "Panel response prediction through reduced order models with application to hypersonic aircraft," *56th AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, 2015, p. 1630.

[16] Howard, M., Bradley, A., Bova, S. W., Overfelt, J., Wagnild, R., Dinzl, D., Hoemmen, M., and Klinvex, A., "Towards Performance Portability in a Compressible CFD Code," *23rd AIAA Computational Fluid Dynamics Conference*, Vol. 1, AIAA, Denver, CO, 2017. doi:10.2514/6.2017-4407, URL https://arc.aiaa.org/doi/abs/10.2514/6.2017-4407.

[17] LeVeque, R. J., *Finite volume methods for hyperbolic problems*, Vol. 31, Cambridge university press, 2002.

[18] Eymard, R., Gallouët, T., and Herbin, R., "Finite volume methods," *Handbook of numerical analysis*, Vol. 7, 2000, pp. 713–1018.

[19] Holmes, P., Lumley, J., and Berkooz, G., *Turbulence, Coherent Structures, Dynamical Systems and Symmetry*, Cambridge University Press, 1996.

[20] Prud'homme, C., Rovas, D., Veroy, K., Machiels, L., Maday, Y., Patera, A., and Turinici, G., "Reliable real-time solution of parameterized partial differential equations: Reduced-basis output bound methods," *Journal of Fluids Engineering*, Vol. 124, No. 1, 2002, pp. 70–80.

[21] Rozza, G., Huynh, D. B. P., and Patera, A. T., "Reduced basis approximation and a posteriori error estimation for affinely parametrized elliptic coercive partial differential equations," *Archives of Computational Methods in Engineering*, Vol. 15, No. 3, 2008, pp. 229–275.

[22] Lieu, T., and Lesoinne, M., "Parameter adaptation of reduced order models for three-dimensional flutter analysis," *42nd AIAA Aerospace Sciences Meeting and Exhibit*, 2004, p. 888.

[23] Amsallem, D., and Farhat, C., "An online method for interpolating linear parametric reduced-order models," *SIAM Journal on Scientific Computing*, Vol. 33, No. 5, 2011, pp. 2169–2198.

[24] Panzer, H., Mohring, J., Eid, R., and Lohmann, B., "Parametric model order reduction by matrix interpolation," *at-Automatisierungstechnik Methoden und Anwendungen der Steuerungs-, Regelungs-und Informationstechnik*, Vol. 58, No. 8, 2010, pp. 475–484.

[25] Degroote, J., Vierendeels, J., and Willcox, K., "Interpolation among reduced-order matrices to obtain parameterized models for design, optimization and probabilistic analysis," *International Journal for Numerical Methods in Fluids*, Vol. 63, No. 2, 2010, pp. 207–230.

[26] Washabaugh, K. M., Zahr, M. J., and Farhat, C., "On the use of discrete nonlinear reduced-order models for the prediction of steady-state flows past parametrically deformed complex geometries," *54th AIAA Aerospace Sciences Meeting*, 2016, p. 1814.

[27] Barber, C. B., Dobkin, D. P., Dobkin, D. P., and Huhdanpaa, H., "The quickhull algorithm for convex hulls," *ACM Transactions on Mathematical Software (TOMS)*, Vol. 22, No. 4, 1996, pp. 469–483.

[28] Alfeld, P., "A trivariate clough—tocher scheme for tetrahedral data," *Computer Aided Geometric Design*, Vol. 1, No. 2, 1984, pp. 169–181.

[29] Renka, R. J., Renka, R., and CLINE, A., "A triangle-based C^1 interpolation method," *The Rocky Mountain journal of mathematics*, 1984, pp. 223–237.

[30] Schaback, R., "Error estimates and condition numbers for radial basis function interpolation," *Advances in Computational Mathematics*, Vol. 3, No. 3, 1995, pp. 251–264.

[31] Harder, R. L., and Desmarais, R. N., "Interpolation using surface splines." *Journal of aircraft*, Vol. 9, No. 2, 1972, pp. 189–191.

[32] Beatson, R. K., and Light, W. A., "Fast evaluation of radial basis functions: methods for two-dimensional polyharmonic splines," *IMA Journal of Numerical Analysis*, Vol. 17, No. 3, 1997, pp. 343–372.

[33] Howard, M., Bradley, A., Bova, S. W., Overfelt, J., Wagnild, R., Dinzl, D., Hoemmen, M., and Klinvex, A., "Towards performance portability in a compressible cfd code," *23rd AIAA Computational Fluid Dynamics Conference*, 2017, p. 4407.

[34] Rizzi, F., Blonigan, P. J., and Carlberg, K. T., "Pressio: Enabling projection-based model reduction for large-scale nonlinear dynamical systems," *arXiv preprint arXiv:2003.07798*, 2020.

[35] Dolvin, D., "Hypersonic international flight research and experimentation (HIFiRE) fundamental science and technology development strategy," *15th AIAA International Space Planes and Hypersonic Systems and Technologies Conference*, 2008, p. 2581.

[36] Wadhams, T., Mundy, E., MacLean, M., and Holden, M., "Ground test studies of the HIFiRE-1 transition experiment part 1: experimental results," *Journal of Spacecraft and Rockets*, Vol. 45, No. 6, 2008, pp. 1134–1148.

[37] Spalart, P., and Allmaras, S., "A one-equation turbulence model for aerodynamic flows," *30th aerospace sciences meeting and exhibit*, 1992, p. 439.