SAND2020-4925J

# Model reduction for steady hypersonic aerodynamics via conservative manifold least-squares Petrov–Galerkin projection

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*

Kevin T. Carlberg§
*University of Washington, Seattle, WA, 98195*

**High-speed aerospace engineering applications rely heavily on computational fluid dynamics (CFD) models for design and analysis. This reliance on CFD models necessitates performing accurate and reliable uncertainty quantification (UQ) of the CFD models, which can be very expensive for hypersonic flows. Additionally, UQ approaches are "many-query" problems requiring many runs with a wide range of input parameters.**

**One way to enable computationally expensive models to be used in such many-query problems is to employ projection-based reduced-order models (ROMs) in lieu of the (high-fidelity) full-order model. In particular, the least-squares Petrov–Galerkin (LSPG) ROM (equipped with hyper-reduction) has demonstrated the ability to significantly reduce simulation costs while retaining high levels of accuracy on a range of problems including subsonic CFD applications. This allows LSPG ROM simulations to replace the full-order model simulations in UQ studies, making UQ tractable even for large-scale CFD models.**

**This work presents the first application of LSPG to a hypersonic CFD application, the HIFiRE-1 in a three-dimensional, turbulent Mach 7.1 flow. We show the ability of the ROM to significantly reduce computational costs while maintaining high levels of accuracy in computed quantities of interest.**

*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
§Affiliate Associate Professor of Applied Mathematics and Mechanical Engineering

# I. Introduction

Hypersonic aerodynamics plays a crucial role in a range of aerospace engineering applications including the design and analysis of missiles, launch vehicles, and reentry vehicles. The expense and difficulty of flight tests and experiments for hypersonic applications has resulted in greater reliance on computational models for design and analysis than in other flight regimes. This dependence poses the need for uncertainty quantification (UQ) to enable practitioners to study and characterize the sources and propagation of error and uncertainties in these computational frameworks [1–4].

Virtually all UQ approaches are "many-query" because they require many evaluations of the model of interest. Hence, if the system of interest is computationally expensive to query, UQ studies can become intractable. This is the case for hypersonic aerodynamics models, which often associate with finite-volume (FV) computational fluid dynamics (CFD) models characterized by highly nonlinear behavior and a large number of conserved variables when non-equilibrium thermochemical effects are included. Surrogate and reduced-order models (ROMs) are thus necessary to overcome this barrier, and enable UQ for problems in hypersonic aerodynamics where the exploration of a variety of parameters and operating conditions is key to characterize the response of a system.

A number of studies have been conducted by applying surrogate and low-fidelity approaches to hypersonic aerodynamics models. Many of these proposed aerodynamic surrogates are low-fidelity models arising from simplified physics, including two-dimensional oblique shock relations [5] and piston-theory aerodynamics [6]. Crowell and McNamara[7] applied a hybrid approach, which computed the steady load components using proper orthogonal decomposition (POD) followed by a kriging interpolant of the POD coefficients in the input-parameter space,* and computed the unsteady load via an analytical correction . It should be noted that while Galerkin-projection ROMs have been applied to both linear [8] and nonlinear [9] finite-element heat-transfer models of hypersonic vehicles, they have not yet been applied to hypersonic CFD to our knowledge.

Projection-based ROMs can potentially provide an improvement in accuracy and robustness over the simplified-physics and POD coefficient-interpolation approaches previously used for reducing the cost of hypersonic aerodynamics simulations. This is due to the fact that projection-based ROMs remain strongly 'tied' to the high-fidelity physics, as they achieve computational savings by applying a projection process directly to the equations governing the high-fidelity model.

In this work, we focus on the least-squares Petrov-Galerkin (LSPG) projection [10], due to its observed accuracy and stability on large-scale problems in CFD [11, 12], and its flexible optimization-based formulation that readily admits integration of constraints that enforce conservation laws over subdomains [13]. Specifically, we present the application of LSPG to hypersonic CFD simulations. To our knowledge, this is the first application of projection-based ROMs to hypersonic CFD simulations.

*Note that this approach is not a projection-based ROM, as it computes POD coefficients using interpolation, not projection of the governing equations.

The paper begins with an overview of the full-order model in section II, followed by an overview of projection-based ROM techniques in section III, and a series of numerical experiments with a CFD simulation of a HIFiRE-1 wind tunnel test in section IV. Finally, section V offers some conclusions and directions for future work.

## II. Full-order model: finite-volume discretization of hypersonic aerodynamic flows

This work considers parameterized systems of *physical conservation laws*. In integral form, the governing equations correspond to

$$\frac{d}{dt}\int_{\omega} u_i(\vec{x},t;\boldsymbol{\mu})\,\mathrm{d}\vec{x} + \int_{\gamma} \boldsymbol{g}_i(\vec{x},t;\boldsymbol{\mu})\cdot\boldsymbol{n}(\vec{x})\,\mathrm{d}\vec{s}(\vec{x}) = \int_{\omega} s_i(\vec{x},t;\boldsymbol{\mu})\,\mathrm{d}\vec{x}, \quad i\in\mathbb{N}(n_u),\ \forall\omega\subseteq\Omega, \tag{1}$$

which is solved over the time domain $t\in[0,T]$ 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 $\mathbb{N}(a) := \{1,\dots,a\}$. Here, $\omega$ 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 $\omega$, while $\Gamma := \partial\Omega$ denotes the boundary of the domain $\Omega$; $\mathrm{d}\vec{s}(\vec{x})$ denotes integration with respect to the boundary; and $u_i\in\mathbb{R}$, $\boldsymbol{g}_i\in\mathbb{R}^d$, and $s_i\in\mathbb{R}$, $i\in\mathbb{N}(n_u)$ denote the $i$th conserved variable (per unit volume), the flux associated with the $i$th conserved variable (per unit area per unit time), and the source associated with the $i$th conserved variable (per unit volume per unit time), respectively. Finally, $\boldsymbol{n}\in\mathbb{R}^d$ denotes the outward unit normal to $\omega$. We emphasize that equations (1) describe conservation of *any* set of variables $u_i$, $i\in\mathbb{N}(n_u)$, given their respective flux $\boldsymbol{g}_i$ and source $s_i$ functions.

### A. Finite-volume discretization

To discretize the governing equations (1), we apply the finite-volume method [14, 15], as it explicitly enforces conservation over prescribed control volumes. In particular, we assume that the spatial domain $\Omega$ has been partitioned into a mesh $\mathcal{M}$, of $N_\Omega\in\mathbb{N}$ 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 $i$th control volume by $\Gamma_i := \partial\Omega_i$. The $i$th control-volume boundary is partitioned into a set of faces denoted by $\mathcal{E}_i$ such that $\Gamma_i = \{\vec{x}\,|\,\vec{x}\in e,\ \forall e\in\mathcal{E}_i,\ i\in\mathbb{N}(|\mathcal{E}_i|)\}$. Then the full set of $N_e$ faces within the mesh is $\mathcal{E}\equiv\{e_i\}_{i=1}^{N_e} := \cup_{i=1}^{N_\Omega}\mathcal{E}_i$. Enforcing conservation (1) on each control volume in the mesh yields

$$\frac{d}{dt}\int_{\Omega_j} u_i(\vec{x},t;\boldsymbol{\mu})\,\mathrm{d}\vec{x} + \int_{\Gamma_j} \boldsymbol{g}_i(\vec{x},t;\boldsymbol{\mu})\cdot\boldsymbol{n}_j(\vec{x})\,\mathrm{d}\vec{s}(\vec{x}) = \int_{\Omega_j} s_i(\vec{x},t;\boldsymbol{\mu})\,\mathrm{d}\vec{x}, \quad i\in\mathbb{N}(n_u),\ j\in\mathbb{N}(N_\Omega), \tag{2}$$

where $\boldsymbol{n}_j \in \mathbb{R}^d$ denotes the unit normal to control volume $\Omega_j$. Finite-volume schemes complete the spatial discretization by forming a state vector $\boldsymbol{x} \in \mathbb{R}^N$ with $N = N_\Omega n_u$ such that

$$x_{\mathcal{I}(i,j)}(t;\boldsymbol{\mu}) = \frac{1}{|\Omega_j|}\int_{\Omega_j} u_i(\vec{x},t;\boldsymbol{\mu})\,\mathrm{d}\vec{x}, \quad i \in \mathbb{N}(n_u),\ j \in \mathbb{N}(N_\Omega), \tag{3}$$

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, and a velocity vector $\boldsymbol{f}(\boldsymbol{w},\tau;\boldsymbol{\nu}) = \boldsymbol{f}^g(\boldsymbol{w},\tau;\boldsymbol{\nu}) + \boldsymbol{f}^s(\boldsymbol{w},\tau;\boldsymbol{\nu})$ 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 $i$th conserved variable (per unit area per unit time). Substituting $\int_{\Omega_j} u_i(\vec{x},t;\boldsymbol{\mu})\,\mathrm{d}\vec{x} \leftarrow |\Omega_j| x_{\mathcal{I}(i,j)}(t;\boldsymbol{\mu})$, $\boldsymbol{g}_i \leftarrow \boldsymbol{g}_i^{\mathrm{FV}}$, and $s_i \leftarrow s_i^{\mathrm{FV}}$ in Eq. (2) and dividing by $|\Omega_j|$ yields

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x},t;\boldsymbol{\mu}), \qquad \boldsymbol{x}(0;\boldsymbol{\mu}) = \boldsymbol{x}^0(\boldsymbol{\mu}), \tag{4}$$

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. (4) as the FOM ODE.

In the case of a steady-state solution, the velocity exhibits no time dependence such that the FOM ODE (4) reduces to

$$\boldsymbol{f}(\boldsymbol{x};\boldsymbol{\mu}) = \boldsymbol{0}, \tag{5}$$

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

### B. Hypersonic aerodynamics

In this paper, we consider high-Mach external aerodynamics. We consider flows with enthalpy not sufficiently high to drive dissociation of the gas we consider (air). Hence, we solve the perfect gas, compressible Navier–Stokes equations. The governing equations, equation of state, transport properties and boundary conditions are presented in Appendix V.A.

### C. Computational barrier: many-query problems

For hypersonic aerodynamics models it is vital to estimate uncertainty to design robust flight vehicles and to determine robust control policies. There are many sources of uncertainty characterizing hypersonic CFD simulations,

including flight conditions, vehicle geometry deformation, turbulence model parameters, turbulence model form, and boundary layer transition location. Although the examples presented in this work do not model chemical non-equilibrium effects, it should be noted that non-equilibrium chemistry models contain many additional uncertain parameters. To obtain reasonable estimates of uncertainty from a UQ approach applied to hypersonic aerodynamics, due to the large number of uncertain parameters, $n_\mu$, it is necessary to evaluate the model many times for many different realizations of parameters $\boldsymbol{\mu}$. In addition, hypersonic CFD can require a large computational mesh, $\mathcal{M}$, to sufficiently resolve shockwaves and the large temperature and velocity gradients near surfaces. This can yield a large state-space dimension $N$ (e.g., $N \sim 10^7$).

This introduces a *de facto* computational barrier: the full-order model is too computationally expensive to solve enough times to obtain reasonable uncertainty estimates. Such cases demand a method for *approximately* solving the full-order model while retaining high levels of accuracy. We now present a method that (a) computes a low-dimensional representation of the state using a linear subspace, and (b) computes a dynamics model for the resulting latent state that exactly satisfies the physical conservation laws over *subdomains* comprising unions of control volumes of the mesh.

## III. Reduced-order modeling

### A. Least-squares Petrov–Galerkin projection

Classical projection-based reduced-order models compute an approximate solution $\tilde{\boldsymbol{x}} \approx \boldsymbol{x}$ from an affine function

$$\tilde{\boldsymbol{x}}(t;\boldsymbol{\mu}) = \boldsymbol{x}^0(\boldsymbol{\mu}) + \boldsymbol{\Phi}\hat{\boldsymbol{x}}(t;\boldsymbol{\mu}), \tag{6}$$

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. This basis can be computed in a variety of ways during the offline stage, e.g., eigenmode analysis, POD [16], or the reduced-basis method [17, 18]. Typically, $\boldsymbol{\Phi}$ is orthonormal, and $\boldsymbol{\Phi}^T\boldsymbol{\Phi} = \boldsymbol{I}$. Note that it does not need to be orthonormal for LSPG, unlike other ROM methods such as Galerkin projection.

In the case of a steady simulation, LSPG substitutes the approximation $\boldsymbol{x} \leftarrow \tilde{\boldsymbol{x}}$ into the FOM steady-state equations (5), and subsequently minimizes residual in a weighted $\ell^2$-norm, i.e.,

$$\hat{\boldsymbol{x}} = \underset{\hat{\boldsymbol{z}}\in\mathbb{R}^p}{\arg\min} \|\boldsymbol{A}\boldsymbol{f}(\boldsymbol{x}^0(\boldsymbol{\mu}) + \boldsymbol{\Phi}\hat{\boldsymbol{z}};\boldsymbol{\mu})\|_2. \tag{7}$$

where $\boldsymbol{A} = \boldsymbol{I}$, for example. However, to ensure that this model incurs an $N$-independent operation count, this weighting matrix should be sparse in the sense that it has a small number of nonzero columns. In this case, one can set $\boldsymbol{A} = (\boldsymbol{P}_r\boldsymbol{\Phi}_r)^+\boldsymbol{P}_r$ and $\boldsymbol{A} = \boldsymbol{P}_r$ in the case of gappy POD and collocation, respectively. Here, $\boldsymbol{P}_r \in \{0,1\}^{n_{p,r}\times N}$ denotes a sampling matrix comprising selected rows of the $N \times N$ identity matrix, while $\boldsymbol{\Phi}_r \in \mathbb{R}^{N\times p_r}$ is a $p_r$-dimensional

reduced-basis matrix constructed for the residual $\boldsymbol{f}$. Employing the gappy POD approximation results in the GNAT reduced-order model [11].

### B. Conservative LSPG projection

As proposed in Ref. [13], we now modify the LSPG ROM for steady simulations (Eq. (7)) by enforcing conservation on the decomposed mesh as nonlinear equality constraints. In particular, conservative LSPG (C-LSPG) projection for steady simulations computes a solution $\hat{\boldsymbol{x}}$ that satisfies

$$\begin{aligned} &\underset{\hat{\boldsymbol{z}}\in\mathbb{R}^p}{\text{minimize}} \ \|\boldsymbol{A}\boldsymbol{f}(\boldsymbol{x}^0(\boldsymbol{\mu}) + \boldsymbol{\Phi}\hat{\boldsymbol{z}};\boldsymbol{\mu})\|_2 \\ &\text{subject to } \bar{\boldsymbol{C}}\boldsymbol{f}(\boldsymbol{x};\boldsymbol{\mu}) = \boldsymbol{0}. \end{aligned} \tag{8}$$

where $\bar{\boldsymbol{C}}$ will be defined in section III.C.

For hyper-reduced C-LSPG, we can instead satisfy an approximate conservation constraint

$$\begin{aligned} &\underset{\hat{\boldsymbol{z}}\in\mathbb{R}^p}{\text{minimize}} \ \|\boldsymbol{A}\boldsymbol{f}(\boldsymbol{x}^0(\boldsymbol{\mu}) + \boldsymbol{\Phi}\hat{\boldsymbol{z}};\boldsymbol{\mu})\|_2 \\ &\text{subject to } \bar{\boldsymbol{C}}\tilde{\tilde{\boldsymbol{f}}}(\boldsymbol{x};\boldsymbol{\mu}) = \boldsymbol{0}. \end{aligned} \tag{9}$$

where $\tilde{\tilde{\boldsymbol{f}}}$ is the approximate residual vector, which is $\tilde{\tilde{\boldsymbol{f}}} = \boldsymbol{P}_r^T\boldsymbol{P}_r\boldsymbol{f}$ and $\tilde{\tilde{\boldsymbol{f}}} = \boldsymbol{\Phi}_r(\boldsymbol{P}_r\boldsymbol{\Phi}_r)^+\boldsymbol{P}_r\boldsymbol{f}$ in the case of collocation and gappy POD, respectively. Ref. [13] contains additional details on conservative LSPG projection, including sufficient conditions for feasibility of the associated optimization problems, and *a posteriori* error bounds.

### C. Conservation constraints

To begin, we decompose the mesh $\mathcal{M}$ into subdomains, each of which comprises the union of control volumes. That is, we define a decomposed mesh $\bar{\mathcal{M}}$ of $N_{\bar{\Omega}}(\leq N_\Omega)$ subdomains $\bar{\Omega}_i = \cup_{j\in\mathcal{K}\subseteq\mathbb{N}(N_\Omega)}\Omega_j$, $i \in \mathbb{N}(N_{\bar{\Omega}})$ with $\bar{\mathcal{M}} := \{\bar{\Omega}_i\}_{i=1}^{N_{\bar{\Omega}}}$. Denoting the boundary of the $i$th subdomain by $\bar{\Gamma}_i := \partial\bar{\Omega}_i$, we have $\bar{\Gamma}_i = \{\vec{x} \,|\, \vec{x} \in e, \ \forall e \in \bar{\mathcal{E}}_i, \ i \in \mathbb{N}(|\bar{\mathcal{E}}_i|)\} \subseteq \cup_{j=1}^{N_\Omega}\Gamma_j$, $i \in \mathbb{N}(N_{\bar{\Omega}})$ with $\bar{\mathcal{E}}_i \subseteq \mathcal{E}$ representing the set of faces belonging to the $i$th subdomain. We denote the full set of faces within the decomposed mesh by $\bar{\mathcal{E}} := \cup_{i=1}^{N_{\bar{\Omega}}}\bar{\mathcal{E}}_i \subseteq \mathcal{E}$. Note that the global domain can be considered by employing $\bar{\mathcal{M}} = \bar{\mathcal{M}}_{\text{global}}$, which is characterized by $N_{\bar{\Omega}} = 1$ subdomain that corresponds to the global domain.

Enforcing conservation (1) on each subdomain in the decomposed mesh yields

$$\int_{\bar{\Gamma}_j} \boldsymbol{g}_i(\vec{x},t;\boldsymbol{\mu})\cdot\bar{\boldsymbol{n}}_j(\vec{x})\,\mathrm{d}\vec{s}(\vec{x}) = \int_{\bar{\Omega}_j} s_i(\vec{x},t;\boldsymbol{\mu})\,\mathrm{d}\vec{x}, \quad i \in \mathbb{N}(n_u), \ j \in \mathbb{N}(N_{\bar{\Omega}}), \tag{10}$$

where $\bar{\boldsymbol{n}}_j \in \mathbb{R}^d$ denotes the unit normal to subdomain $\bar{\Omega}_j$. We propose applying the same finite-volume discretization employed to discretize the control-volume conservation equations (2) to the subdomain conservation equations (10). To

accomplish this, we introduce a "decomposed" state vector $\bar{\boldsymbol{x}} \in \mathbb{R}^{\bar{N}}$ with $\bar{N} = N_{\bar{\Omega}} n_u$ and elements

$$\bar{x}_{\bar{I}(i,j)}(\boldsymbol{x};\boldsymbol{\mu}) = \frac{1}{|\bar{\Omega}_j|} \int_{\bar{\Omega}_j} u_i(\vec{x};\boldsymbol{\mu})\, \mathrm{d}\vec{x}, \quad i \in \mathbb{N}(n_u),\ j \in \mathbb{N}(N_{\bar{\Omega}}), \tag{11}$$

where $\bar{I} : \mathbb{N}(n_u) \times \mathbb{N}(N_{\bar{\Omega}}) \to \mathbb{N}(\bar{N})$ denotes a mapping from conservation-law index and subdomain index to decomposed degree of freedom. The decomposed state vector can be computed from the state vector $\boldsymbol{x}$ as

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

where $\bar{\boldsymbol{C}} \in \mathbb{R}_{+}^{\bar{N} \times N}$ has elements

$$\bar{\boldsymbol{C}}_{\bar{I}(i,j),I(l,k)} = \frac{|\Omega_k|}{|\bar{\Omega}_j|} \delta_{il} I(\Omega_k \subseteq \bar{\Omega}_j), \tag{12}$$

where $I$ is the indicator function, which evaluates to one if its argument is true, and zero if its argument is false.

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

$$\bar{\boldsymbol{f}}(\boldsymbol{x};\boldsymbol{\mu}) = \bar{\boldsymbol{C}}\boldsymbol{f}(\boldsymbol{x};\boldsymbol{\mu}), \tag{13}$$

such that subdomain conservation can be expressed as

$$\bar{\boldsymbol{C}}\boldsymbol{f}(\boldsymbol{x};\boldsymbol{\mu}) = 0. \tag{14}$$

For the detailed explanation on the derivation of Eqs. (13)–(14), we refer readers Ref. [13, Section 4.1].

### D. Manifold Least-squares Petrov–Galerkin projection

One of the shortcomings of the affine linear trial subspace used to compute the approximate state $\tilde{\boldsymbol{x}}$ in (6) is that there exists some $\hat{\boldsymbol{x}}$ such that $\tilde{\boldsymbol{x}}$ will contain some non-physical local phenomena such as regions of negative density or temperature. Ref [19, Section VI.E] shows that ensuring $\tilde{\boldsymbol{x}}$ does not have non-physical local flow features can significantly improve the robustness of LSPG and Galerkin ROMs. In this paper, we propose and demonstrate a nonlinear *trial manifold* that approximates full-order model states without non-physical local features. We consider $\tilde{\tilde{\boldsymbol{x}}} \approx \boldsymbol{x}$ of the form

$$\tilde{\tilde{\boldsymbol{x}}}(t;\boldsymbol{\mu}) = \boldsymbol{h}(\boldsymbol{x}^0(\boldsymbol{\mu}) + \boldsymbol{\Phi}\hat{\boldsymbol{x}}(t;\boldsymbol{\mu})), \tag{15}$$

where $\tilde{\tilde{\boldsymbol{x}}} \in \mathcal{S}$ and $\mathcal{S} := \{\boldsymbol{h}(\boldsymbol{x}^0(\boldsymbol{\mu}) + \boldsymbol{\Phi}\hat{\boldsymbol{z}}) \,|\, \hat{\boldsymbol{z}} \in \mathbb{R}^p\}$ denotes the nonlinear trial manifold from the extrinsic view. Here $\boldsymbol{h}(\tilde{\boldsymbol{x}}) \in \mathbb{R}^N$ denotes the clipping function, which comprises a nonlinear mapping from a potentially non-physical linear

affine subspace $\hat{\boldsymbol{x}} \in \mathbb{R}^p$ to a manifold on which quantities like density and temperature only take physical, non-negative values.

Similarly to section III.A, steady manifold LSPG (M-LSPG) substitutes the approximation $\boldsymbol{x} \leftarrow \tilde{\tilde{\boldsymbol{x}}}$ into the FOM steady-state equations (5), and subsequently minimizes residual in a weighted $\ell^2$-norm, i.e., [20]

$$\hat{\boldsymbol{x}}(\boldsymbol{\mu}) = \underset{\hat{\boldsymbol{z}} \in \mathbb{R}^p}{\arg\min} \left\| \boldsymbol{A}\boldsymbol{f}\left(\boldsymbol{h}(\boldsymbol{x}^0(\boldsymbol{\mu}) + \boldsymbol{\Phi}\hat{\boldsymbol{z}}(\boldsymbol{\mu})); \boldsymbol{\mu}\right)\right\|_2, \tag{16}$$

As for LSPG in (8), constraints can be applied to the M-LSPG minimization statement (16), resulting in conservative manifold LSPG (CM-LSPG). Hyper-reduced constraints can also be applied to CM-LSPG as for C-LSPG in (9).

We choose the clipping function $\boldsymbol{h}(\tilde{\boldsymbol{x}})$ to enforce $\rho > 0$ and $T > 0$ in the flowfield represented by $\tilde{\tilde{\boldsymbol{x}}}$. The density field $\tilde{\tilde{u}}_1$ is computed by

$$\tilde{\tilde{u}}_1 = \max(\epsilon_1, \tilde{u}_1), \tag{17}$$

where $\epsilon_1 > 0$ is some number that should be very small relative to the free stream density and $\tilde{u_1}$ is the density field from $\tilde{\boldsymbol{x}}$. The expression $T > 0$ can be written in terms of the conserved quantities in (26) using (27) and (35) to derive the following expression for temperature

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

Since $c_v > 0, T > 0$ can be enforced in $\tilde{\tilde{\boldsymbol{x}}}$ by setting

$$\tilde{\tilde{u}}_5 = \max\left(\epsilon_5 + \frac{1}{2\tilde{\tilde{u}}_1}\left[\tilde{u}_2^2 + \tilde{u}_3^2 + \tilde{u}_4^2\right], \tilde{u}_5\right), \tag{18}$$

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 (17) to $\tilde{\boldsymbol{x}}$, and $\tilde{\tilde{\boldsymbol{x}}} = \boldsymbol{h}_5(\tilde{z})$ applies (18) to $\tilde{z}$. Note that this is similar to the compositions of nonlinear activation functions that make up neural nets, so this idea could be extended to the autoencoder neural net architectures used as manifold approximations in [20].

## IV. Numerical experiments

The results presented below have been obtained using two codes being developed at Sandia National Laboratories, namely SPARC and *Pressio*[†]. SPARC (Sandia Parallel Aerodynamics and Reentry Code) is a compressible CFD code focused on aerodynamics and aerothermodynamics problems. It solves the compressible Navier–Stokes and Reynolds-Averaged Navier–Stokes (RANS) equations on structured and unstructured grids using a cell-centered finite

[†]https://github.com/Pressio

**Table 1 Free stream flow conditions for the HiFIRE-1 for run 34 of the CUBRC wind tunnel experiments [23].**

| | |
|---|---|
| Density | 0.070215 $kg/m^3$ |
| Velocity | 2168.7 $m/s$ |
| Mach Number | 7.1 |
| Angle of attack | 2.0° |
| Temperature | 231.91 $K$ |
| Reynolds Number | 10,000,000 $1/m$ |

volume discretization scheme [21]. Its target use cases are transonic flows to support gravity bomb analyses and hypersonic flows for re-entry vehicle analyses. SPARC also solves the transient heat equation and associated equations for non-decomposing and decomposing ablators on unstructured grids using a Galerkin finite element method. One and two-way multiphysics couplings exist between the CFD and ablation solvers within the code.

*Pressio* is an open-source C++11 header-only library aimed at enabling parallel, scalable, and performant ROM capabilities to be adopted by any C++ application with a minimally intrusive API [22]. The main design principle behind *Pressio* is that an application only needs to expose, for a given state $\boldsymbol{x}$, time $t$, and parameters $\boldsymbol{\mu}$, the velocity vector $\boldsymbol{f}(\boldsymbol{x},t;\boldsymbol{\mu})$ and the action of the Jacobian matrix $\partial \boldsymbol{f}(\boldsymbol{x},t;\boldsymbol{\mu})/\partial \boldsymbol{x}$. Using C++ metaprogramming, *Pressio* detects and leverages the application's native data structures (e.g., vector, matrix) to instantiate and run the desired ROM methods. A compile-time check is performed by *Pressio* to verify if the target application satisfies the correct API, and if not, a compile time error is thrown. Exposing from SPARC the required functionalities was relatively easy, since it involved the creation of a new adapter class and no changes to the original SPARC code. We remark that while in this work we limit our attention to LSPG, the *same* interface developed in SPARC can now be used to run *any* of the ROM methods supported in *Pressio*.

### A. HIFiRE-1

We demonstrate LSPG, C-LSPG, and CM-LSPG on a SPARC simulation of a wind tunnel test of the HiFIRE-1 (Hypersonic International Flight Research Experimentation) vehicle. The baseline case we use in this paper is run 34 of the experimental campaign undertaken at the CALSPAN University of Buffalo Research Center (CUBRC) [23]. The corresponding free stream conditions are listed in table 1. Additionally, turbulent transition is modeled by tripping the boundary layer at x=0.35 m downstream from the leading edge of the vehicle.

### B. Full-Order Model

The HIFiRE-1 outer mold line geometry is axisymmetric. Because the angle of attack for run 34 is non-zero, the vehicle is modeled with the mesh shown in Figure 1, which discretizes half of the flow field and assumes flow symmetry about the center line. The mesh has 2,031,616 cells, corresponding to a state-space size of 12,189,696 with $n_u = 6$ since we are using the Spalart-Allmaras turbulence model. The flow is solved using pseudo-time-stepping with a backward

**Fig. 1 HIFiRE mesh.**

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 pseudo-time steps. Near the baseline parameters, the solver converges in around 9,000 steps, but convergence is slower at lower values of freestream density and velocity.

Figure 2 shows the flow field at the baseline conditions listed in table 1. Noteworthy off-body flow features include a bow shock near the nose, an expansion wave at the back end of the nose cone, and an oblique shock wave upstream of the flare. Boundary layer transition is visible in the sudden increase in wall heat flux downstream of the leading edge. Note that $y^+$ is at most 1.0 for the baseline case, which implies sufficient resolution of the near wall portion of the boundary layer for steady RANS equations. Additionally, the range of flow conditions used in this paper were chosen to keep the Reynolds number below that of the baseline case, and to keep the leading shockwave from touching the inflow boundary of the computational domain.

Validation studies have found that the full order model computes heat fluxes similar to those observed in the wind tunnel experiments on most of the vehicle, but underpredicts it at the beginning of the flare [24]. This is a difficult region to predict because flow separation at the cylinder-flare intersection is highly sensitive and is not well modeled by RANS turbulence models.

### C. Reduced-Order Model

To compute the POD modes for the ROM, we first assemble the snapshot matrix $\boldsymbol{X}_{\text{snap}}$ where the $i$th column corresponds to a FOM solution $\boldsymbol{x}^i$ at parameter values $\boldsymbol{\mu}_i$. The reference state $\boldsymbol{x}^0(\boldsymbol{\mu})$ is simply the mean of all snapshots. The basis $\boldsymbol{\Phi}$ is then defined as

$$\boldsymbol{\Phi} = \boldsymbol{D}_{max}\bar{\boldsymbol{\Phi}}, \tag{19}$$

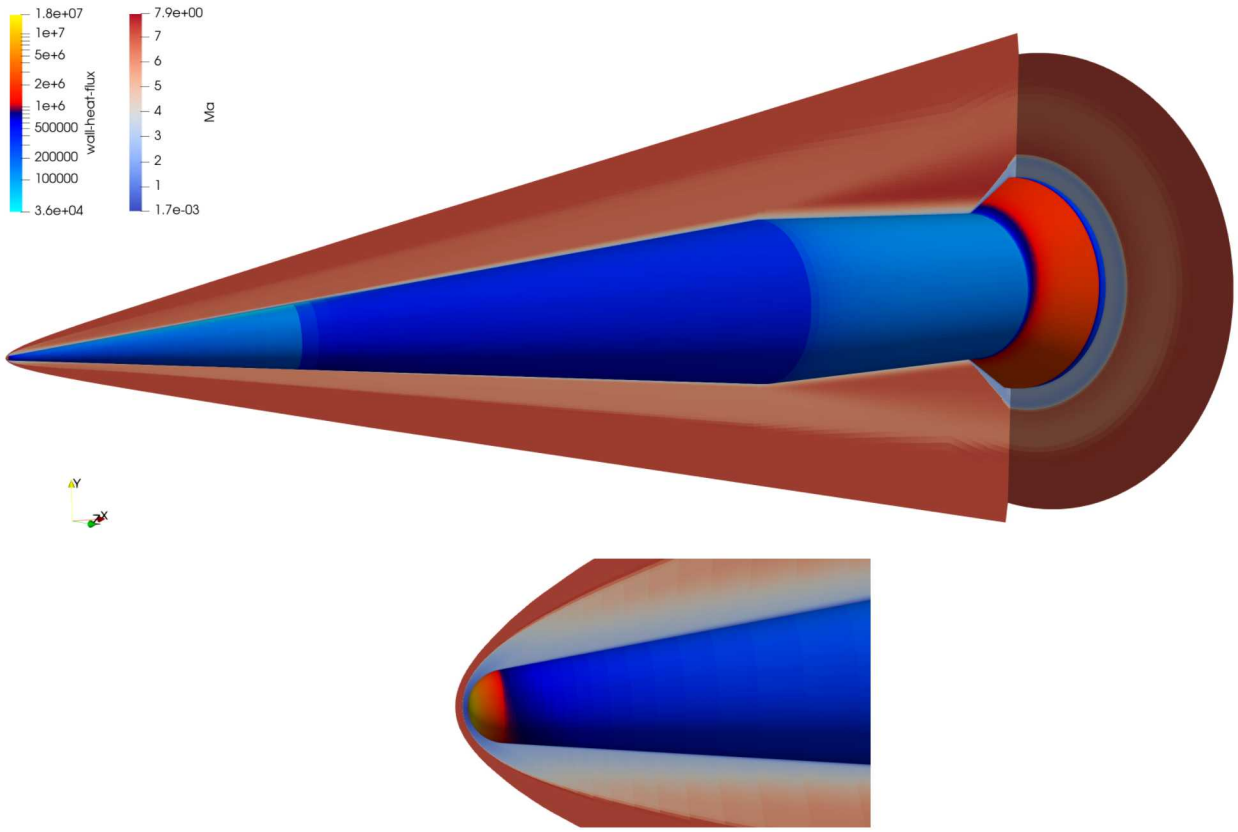



**Fig. 2 SPARC simulation of HIFiRE-1 at the freestream condition in table 1. The flow field is colored by Mach number $M$, and the vehicle surface is colored by wall heat flux $q_{wall}$. The bottom plot shows a magnified view of the tip.**

where $\boldsymbol{D}_{max} \in \mathbb{R}^{n_{\boldsymbol{P}} \times N}$ is a diagonal matrix with the maximum absolute value of each conserved quantity along the main diagonal and $\bar{\boldsymbol{\Phi}}$ are the POD modes computed from the centered snapshot matrix normalized by $\boldsymbol{D}_{max}$,

$$\boldsymbol{D}_{max}^{-1}(\boldsymbol{X}_{\text{snap}} - \boldsymbol{x}^0). \tag{20}$$

This normalization is done because SPARC is run with dimensional quantities, as is standard practice for hypersonic CFD codes, especially for cases with non-equilibrium chemistry effects. Normalizing snapshot data prior to computing POD modes increases numerical robustness. This is because reducing the range of variable scales makes the normalized, centered snapshot matrix (20) better conditioned than the centered, unscaled snapshot matrix $(\boldsymbol{X}_{\text{snap}} - \boldsymbol{x}^0)$.

Because of the diagonal matrix in (19), the basis $\boldsymbol{\Phi}$ is not orthonormal ($\boldsymbol{\Phi}^T\boldsymbol{\Phi} \neq \boldsymbol{I}$), so projections have to be done with the Moore-Penrose pseudo inverse of $\boldsymbol{\Phi}$. Since $\boldsymbol{D}_{max}$ is diagonal we have

$$\hat{\boldsymbol{x}}(\boldsymbol{\mu}) = (\boldsymbol{\Phi})^{+}(\boldsymbol{x}(\boldsymbol{\mu}) - \boldsymbol{x}^0) = \bar{\boldsymbol{\Phi}}^T \boldsymbol{D}_{max}^{-1}(\boldsymbol{x}(\boldsymbol{\mu}) - \boldsymbol{x}^0). \tag{21}$$

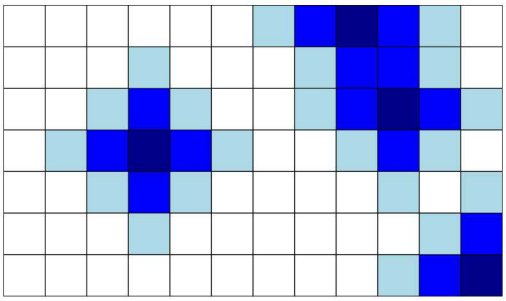


**Fig. 3 Schematic showing cells included in a sample mesh. The residual $f$ is sampled at the dark cells. The neighboring cells (blue), and their neighbors (light blue) are included in the sample mesh since computing $f$ in the middle cell requires the states $x$ at all highlighted cells.**

The selection of the weighting matrix $\boldsymbol{A}$ in (7) is crucial for the accuracy and speed of LSPG. We set $\boldsymbol{A} = \boldsymbol{D} \in \mathbb{R}^{N\times N}$, where $\boldsymbol{D} \in \mathbb{R}^{N\times N}$ is defined as

$$\boldsymbol{D}_{\mathcal{I}_{(i,j)},\mathcal{I}_{(i,j)}} = \Omega_j, \quad i \in \mathbb{N}(n_u), j \in \mathbb{N}(N_\Omega). \tag{22}$$

This is a diagonal matrix whose elements correspond to the size of each control volume $|\Omega_k|$. It is found that this choice of $\boldsymbol{A}$ vastly improves the convergence rate of Gauss–Newton iteration. This choice of $\boldsymbol{A}$ makes the LSPG residual equivalent to that defined for the full order model. One possible reason for the improved convergence observed when applying $\boldsymbol{A}$ is that the increase in relative weighting on the larger cells near the inflow and outflow boundaries of the computational domain improves the accuracy of the ROM upstream of the leading shock. Since the no-slip condition near the wall is the same at all flow conditions, it is implicitly enforced as all POD modes satisfy it. The inflow conditions vary with free stream velocity and density, so these boundary conditions are not automatically satisfied at all parameters. Therefore less relative weight on the smaller near-wall cells will not result in boundary condition violations, while increased weight on the inflow cells will penalize boundary condition violations more heavily there. Similar behavior with regard to residual weighting was observed by [25] for parametric LSPG for steady compressible flows.

Hyper-reduction is done by collocation using $\boldsymbol{A} = \boldsymbol{P}_r\boldsymbol{D} \in \mathbb{R}^{n_{\boldsymbol{P}}\times N}$, as in [25]. The collocation points are chosen randomly, for algorithmic simplicity and a low offline cost relative to other hyper-reduction approaches like GNAT and DEIM. Note that the random cell selection algorithm was designed to ensure that the residual is sampled at each boundary and in each mesh block, as suggested in [10]. Collocation is implemented by use of a sample mesh [11], in which the steady residual $\boldsymbol{f}$ is only computed at the collocation cells. This requires using a mesh that contains state data on all cells required to compute the governing equations. For a second-order finite volume scheme, this means that the sample mesh includes collocation cells, their neighboring cells, and the neighbors of those neighbors, as shown in Figure 3 for a two-dimensional structured mesh.

We consider the LSPG, conservative LSPG (C-LSPG), and conservative manifold LSPG (CM-LSPG) formulations presented in section III. The initial guess for the ROM is computed by an inverse distance interpolation of the projected

training data $\hat{\boldsymbol{x}}^i = (\boldsymbol{\Phi})^+(\boldsymbol{x}^i - \boldsymbol{x}^0)$, specifically [25, Algorithm 23]. The non-linear least-squares problems arising from LSPG is solved via a QR-based Gauss-Newton, while the one stemming from C-LSPG and CM-LSPG is solved via normal equations. The Gauss-Newton solver used is provided by *Pressio*, and is run until the relative residual L2 norm falls below $10^{-5}$ or after 200 iterations. The conservation constraint is applied to the entire mesh, leading to 6 constraints, one for each conserved quantity. For C-LSPG and CM-LSPG, the clipping functions (17) and (18) are applied with $\epsilon_1 = \epsilon_5 = 10^{-6}$. Note that no regions of negative density were found, only negative temperatures had to be clipped implicitly by (18). For C-LSPG and CM-LSPG with hyper-reduction, we only consider the approximate conservation constraint (9). Ref [13] found that using the approximate constraint had very little impact of the accuracy of C-LSPG. Additionally, the approximate conservation constraint is less computationally expensive and easier to implement.

We measure the accuracy of the ROM with the following error metrics. Firstly, the state L2 error, defined as

$$\mathcal{E}_{\boldsymbol{x}} = \frac{\|\boldsymbol{x}(\boldsymbol{\mu}) - \tilde{\boldsymbol{x}}(\boldsymbol{\mu})\|_2}{\|\boldsymbol{x}(\boldsymbol{\mu})\|_2}, \tag{23}$$

where $\boldsymbol{x}(\boldsymbol{\mu})$ and $\tilde{\boldsymbol{x}}(\boldsymbol{\mu})$ are the full state computed with the FOM and some approximation with (6), respectively. The vector $\tilde{\boldsymbol{x}}(\boldsymbol{\mu})$ is usually the ROM solution, but we also compute $\tilde{\boldsymbol{x}}(\boldsymbol{\mu})$ for other states as well. Secondly, we compute the integrated wall heat flux error, defined as

$$\mathcal{E}_{Q_{wall}} = \frac{|Q_{wall}(\boldsymbol{\mu}) - \tilde{Q}_{wall}(\boldsymbol{\mu})|}{|Q_{wall}(\boldsymbol{\mu})|}, \tag{24}$$

where $Q_{wall}(\boldsymbol{\mu})$ and $\tilde{Q}_{wall}(\boldsymbol{\mu})$ are the integrals of heat flux at the wall computed with the FOM and some corresponding approximation with (6), respectively. Integrated wall heat flux was chosen since heating is a key design driver for hypersonic vehicles.

As in [25], errors associated with the ROM initial guess are also presented for both state and heat flux errors. This shows how much additional accuracy the ROM provides over a simple linear inverse-distance interpolation over the basis $\boldsymbol{\Phi}$, which is a form of surrogate model. Therefore, the comparison between the ROM solution and initial guess can be interpreted as a comparison between a ROM and an inexpensive surrogate model.

Finally, the state error is also computed for the projection of the FOM solution on the basis $\boldsymbol{\Phi}$ as

$$\tilde{\boldsymbol{x}}_{FOM}(\boldsymbol{\mu}) \equiv \boldsymbol{\Phi}(\boldsymbol{\Phi})^+(\boldsymbol{x}(\boldsymbol{\mu}) - \boldsymbol{x}^0). \tag{25}$$

This provides a lower bound for the LSPG and C-LSPG ROM state error, since it is the most accurate representation of the FOM solution $\tilde{\boldsymbol{x}}(\boldsymbol{\mu})$ possible with the basis $\boldsymbol{\Phi}$. It is possible for CM-LSPG to obtain lower errors, since the projection is on a nonlinear trial basis that can differ from $\boldsymbol{\Phi}$. Additionally, the lower bound for fields derived from the state vector like $\tilde{\boldsymbol{q}}_{wall}(\boldsymbol{\mu})$ is not necessarily computed from $\tilde{\boldsymbol{x}}_{FOM}(\boldsymbol{\mu})$, so we only compute $\mathcal{E}_{\boldsymbol{x}}$ with $\tilde{\boldsymbol{x}}_{FOM}(\boldsymbol{\mu})$ to

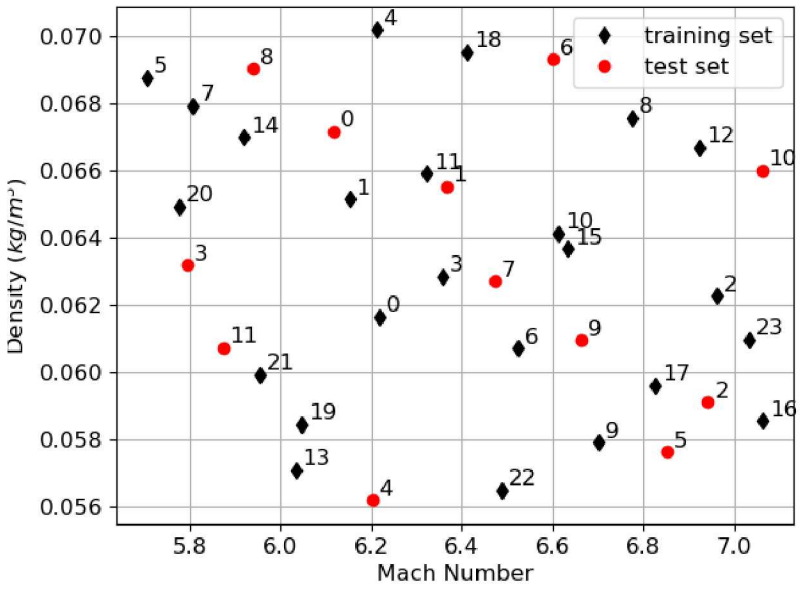



**Fig. 4 Training and test data plotted on parameter space and labeled by case number. The freestream density and velocity were chosen to range from $[0.8\rho_\infty, \rho_\infty]$ and $[0.8v_\infty, v_\infty]$, respectively, where $\rho_\infty$ and $v_\infty$ are the baseline values specified in table 1. The leading shockwave angle changes by roughly $1.5^\circ$ over the range of velocities considered.**

compare with ROM solution and ROM initial guess errors.

Our numerical experiments demonstrate different ROMs parametrized by freestream velocity and density. The training and test data points in the parameter space are selected with a latin hypercube samplings, and shown in Figure 4. The matrices $\boldsymbol{D}_{max}$ and $\bar{\boldsymbol{\Phi}}$ are computed using the FOM solutions at the training points in Figure 4. We consider three different ROM dimensions $p = 1, 2, 4$, using the first 1,2, and 4 basis vectors, respectively. Over 99% of the training set's cumulative statistical energy can be captured with as few as 2 modes. The scales of each conserved quantity vary over 9 orders of magnitude, a very wide range, highlighting the need to scale snapshots prior to computing POD modes for this case. Cumulative energy and conserved quantity scales can be found in appendix V.B.

*1. ROMs without hyper-reduction*

In this section, we discuss the results obtained by running the ROMs *without* hyper-reduction. Figure 4 shows the test set (red markers) where we ran LSPG, C-LSPG and CM-LSPG ROMs, and the corresponding FOM solutions. Table 2 shows state errors, $\mathcal{E}_{\boldsymbol{x}}$, obtained for each ROM. For the sake of clarity, in the table we do not report C-LSPG and CM-LSPG in different columns because they give equivalent results when the clipping functions are inactive. Thus, we use a boldface to highlight the results where the CM-LSPG is used. For $p = 1$, the initial guess has errors ranging from 3 to about 18%, while LSPG, C-LSPG and CM-LSPG offer an improvement, with errors varying between 1 and 4%.

**Table 2 Percent relative L2 error, $100\mathcal{E}_x$, computed for the state vector for different cases and ROM dimensions, $p$. Case numbers correspond to the labels in Figure 4. Errors are labeled as IG for the initial guess and FOM for the error obtained by projecting the FOM solution $x(\mu)$ on the basis $\Phi$ as in (25). Entries are colored by their error value: red for $\mathcal{E}_x > 10\%$, orange for $2\% < \mathcal{E}_x \leq 10\%$, yellow for $1\% < \mathcal{E}_x \leq 2\%$, and green for $\mathcal{E}_x \leq 1\%$. Entries marked as N/A correspond to cases where the ROM computed a non-physical solution and failed. Errors in boldface indicate cases for which the clipping functions are active for some cells. Note that C-LSPG and CM-LSPG give equivalent results when the clipping functions are inactive.**

| | $p = 1$ | | | | $p = 2$ | | | | $p = 4$ | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Case | IG | LSPG | CM-LSPG | FOM | IG | LSPG | CM-LSPG | FOM | IG | LSPG | CM-LSPG | FOM |
| 0 | 3.72 | 2.69 | 2.66 | 2.40 | 3.46 | 2.66 | 0.67 | 0.62 | 3.42 | 2.02 | 0.16 | 0.14 |
| 1 | 3.16 | 1.67 | 1.70 | 1.61 | 3.01 | 7.35 | 1.04 | 1.00 | 2.86 | 1.92 | 0.19 | 0.16 |
| 2 | 4.77 | 3.35 | 3.21 | 2.46 | 4.50 | N/A | **0.67** | 0.66 | 4.45 | 7.10 | 0.12 | 0.11 |
| 3 | 8.06 | 1.44 | 1.51 | 1.37 | 8.15 | 2.90 | 1.35 | 1.30 | 8.05 | 0.47 | 0.12 | 0.11 |
| 4 | 18.35 | 3.61 | 3.67 | 2.95 | 18.72 | 4.12 | 0.88 | 0.84 | 18.70 | 0.49 | 0.20 | 0.19 |
| 5 | 5.00 | 3.99 | 3.85 | 2.90 | 4.57 | N/A | **0.38** | 0.35 | 4.57 | 4.90 | 0.20 | 0.18 |
| 6 | 9.76 | 2.50 | 2.60 | 2.13 | 9.79 | N/A | 0.77 | 0.75 | 9.76 | 2.30 | 0.15 | 0.15 |
| 7 | 3.01 | 1.07 | 1.04 | 0.99 | 3.09 | N/A | 1.02 | 0.99 | 2.95 | 1.86 | 0.17 | 0.15 |
| 8 | 7.04 | 3.74 | 3.67 | 3.23 | 6.92 | 0.68 | 0.61 | 0.61 | 6.90 | 2.28 | 0.34 | 0.28 |
| 9 | 1.40 | 1.84 | 1.72 | 1.36 | 1.00 | N/A | 0.67 | 0.65 | 0.80 | 1.55 | 0.12 | 0.11 |
| 10 | 11.51 | N/A | **1.26** | 1.31 | 11.71 | N/A | **1.29** | 1.31 | 11.64 | 2.51 | 0.22 | 0.19 |
| 11 | 12.29 | 1.07 | 1.01 | 1.00 | 12.38 | 1.63 | 0.79 | 0.77 | 12.37 | 0.30 | 0.20 | 0.18 |

The largest differences in error between the ROMs and the initial guess occur for test cases near the edge of the training set, including cases 4,10, and 11 (see figure 4). We remark that while LSPG and C-LSPG fail for case 10 due to a non-physical initial guess with local regions of negative temperature, CM-LSPG proves to be more robust in that case since it removes the non-physical features.

For a ROM dimension $p = 2$, C-LSPG and CM-LSPG are more accurate and robust than LSPG, which indeed fails for half of the tests cases. Two cases, namely 2 and 10, fail due to non-physical initial guesses that have local regions of negative temperature. The other cases fail when the solver reaches a state with local regions of negative temperature. C-LSPG provides an improvement over LSPG, with only 3 failures, namely cases 2, 5, and 10, which are successfully ran by CM-LSPG. The $\mathcal{E}_x$ for C-LSPG and CM-LSPG is around 1% or less for most cases, which is roughly 3-10 times more accurate than the initial guess.

For ROM dimension $p = 4$, no cases fail for LSPG or C-LSPG solutions. Therefore, the clipping functions are not active, and C-LSPG and CM-LSPG give identical results for this ROM dimension. LSPG offers an improvement in $\mathcal{E}_x$ over the initial guess in most cases, but is less accurate for cases 2,5, and 9. C-LSPG and CM-LSPG have a lower $\mathcal{E}_x$ than the initial guess and no cases fail. For example, for cases 4, 10, and 11, $\mathcal{E}_x$ for the CM-LSPG solution is around 0.2%, about 50-100 times smaller than $\mathcal{E}_x$ for the corresponding initial guess. The C-LSPG/CM-LSPG solution is also around 5-10 times more accurate than the $p = 2$ solutions. The large improvement in accuracy is due to the

**Table 3 Percent relative error, $100\mathcal{E}_{Q_{wall}}$, computed for the integrated wall heat flux for different cases and ROM dimensions, $p$. Labels and abbreviations are defined in table 2.**

| | $p=1$ | | | $p=2$ | | | $p=4$ | | |
|---|---|---|---|---|---|---|---|---|---|
| Case | IG | LSPG | CM-LSPG | IG | LSPG | CM-LSPG | IG | LSPG | CM-LSPG |
| 0 | 0.84 | 3.25 | 3.38 | 0.03 | 0.71 | 1.36 | 0.07 | 2.03 | 1.01 |
| 1 | 0.33 | 2.68 | 2.56 | 1.15 | 3.02 | 1.48 | 1.55 | 1.70 | 0.57 |
| 2 | 0.74 | 1.15 | 1.38 | 1.82 | N/A | **0.42** | 2.10 | 4.13 | 1.02 |
| 3 | 12.58 | 2.16 | 2.54 | 9.41 | 1.31 | 3.03 | 10.05 | 1.55 | 1.39 |
| 4 | 20.80 | 2.93 | 2.78 | 17.48 | 3.10 | 0.95 | 17.46 | 1.80 | 1.85 |
| 5 | 0.91 | 0.56 | 0.81 | 2.37 | N/A | **1.20** | 2.30 | 2.33 | 1.23 |
| 6 | 7.97 | 1.63 | 1.44 | 7.72 | N/A | 0.49 | 8.24 | 2.96 | 1.24 |
| 7 | 2.63 | 1.62 | 1.51 | 1.21 | N/A | 1.59 | 1.79 | 1.81 | 0.55 |
| 8 | 3.65 | 3.49 | 3.73 | 4.14 | 0.87 | 0.80 | 3.75 | 2.45 | 1.28 |
| 9 | 0.41 | 0.66 | 0.44 | 0.52 | N/A | 1.34 | 0.05 | 0.63 | 0.56 |
| 10 | 14.35 | N/A | **3.30** | 10.20 | N/A | **3.11** | 9.35 | 4.44 | 1.87 |
| 11 | 17.00 | 3.24 | 3.66 | 15.16 | 2.36 | 3.19 | 15.56 | 2.08 | 2.24 |

better approximation offered by the $p = 4$ basis, as also shown by the large decrease in FOM projection error when $p$ is increased to 4.

We remark that $\mathcal{E}_{\boldsymbol{x}}$ computed with C-LSPG and/or CM-LSPG is very close to the FOM projection error when $p = 2$ and 4. This means that C-LSPG is nearly as accurate as possible for the basis used. For case 10 with $p = 1$ and $p = 2$, CM-LSPG is actually more accurate than the FOM projection, since the clipping functions were not used when computing the FOM projection. Overall, we can thus conclude that CM-LSPG is the most accurate and robust method for all three ROM sizes, $p$.

Table 3 shows a summary of the results obtained for wall heat flux error, $\mathcal{E}_{Q_{wall}}$, and reveals a different behaviour than the one observed for the $\mathcal{E}_{\boldsymbol{x}}$. For a given run, $\mathcal{E}_{Q_{wall}}$ be considerably larger than $\mathcal{E}_{\boldsymbol{x}}$, and there are more cases for which the ROMs do not offer much or any improvement in accuracy over the initial guess. The improvement in accuracy seems to exhibit a dependence on where the parameter space. Figure 5 shows the ratio between $\mathcal{E}_{Q_{wall}}$ obtained for CM-LSPG and the initial guess error for $p = 4$. This plot shows that the ROMs with the largest improvement in error over the initial guess are all evaluated on the edges of the training data set. Table 3 shows the same behavior for other values of $p$ as well. These results suggest that the ROMs are most useful near the edge of the region spanned by the training parameter set, where surrogate models like the inverse-distance interpolation used to compute the initial guess for the ROM tend to have much larger ($\geq 10\%$) errors.

The ROMs compute $\mathcal{E}_{Q_{wall}}$ relatively accurately for all the test cases considered. For all ROM dimensions, $\mathcal{E}_{Q_{wall}}$ are around 1-4% for CM-LSPG and LSPG. In most cases, $\mathcal{E}_{Q_{wall}}$ is lower for CM-LSPG than LSPG. Additionally, $\mathcal{E}_{Q_{wall}}$ is actually smaller for $p = 2$ than $p = 4$ in some cases. In several cases, CM-LSPG computes a lower value

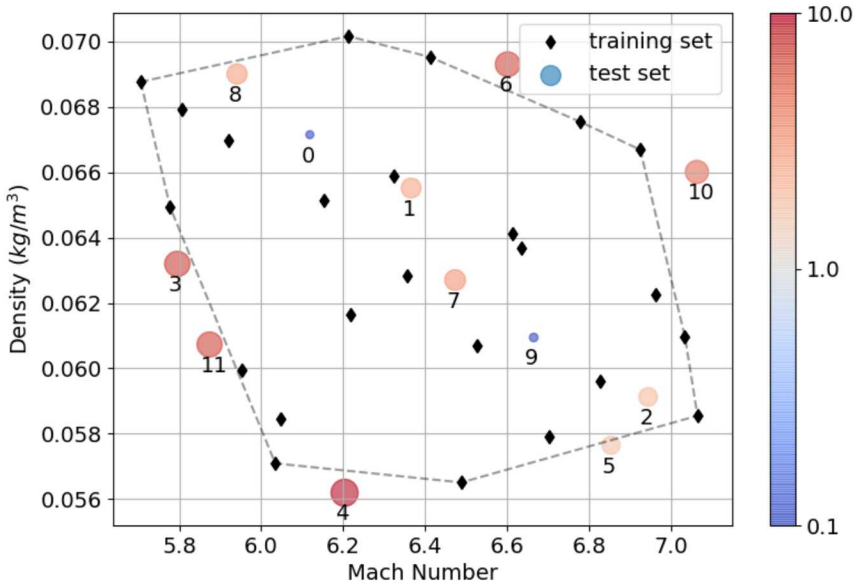


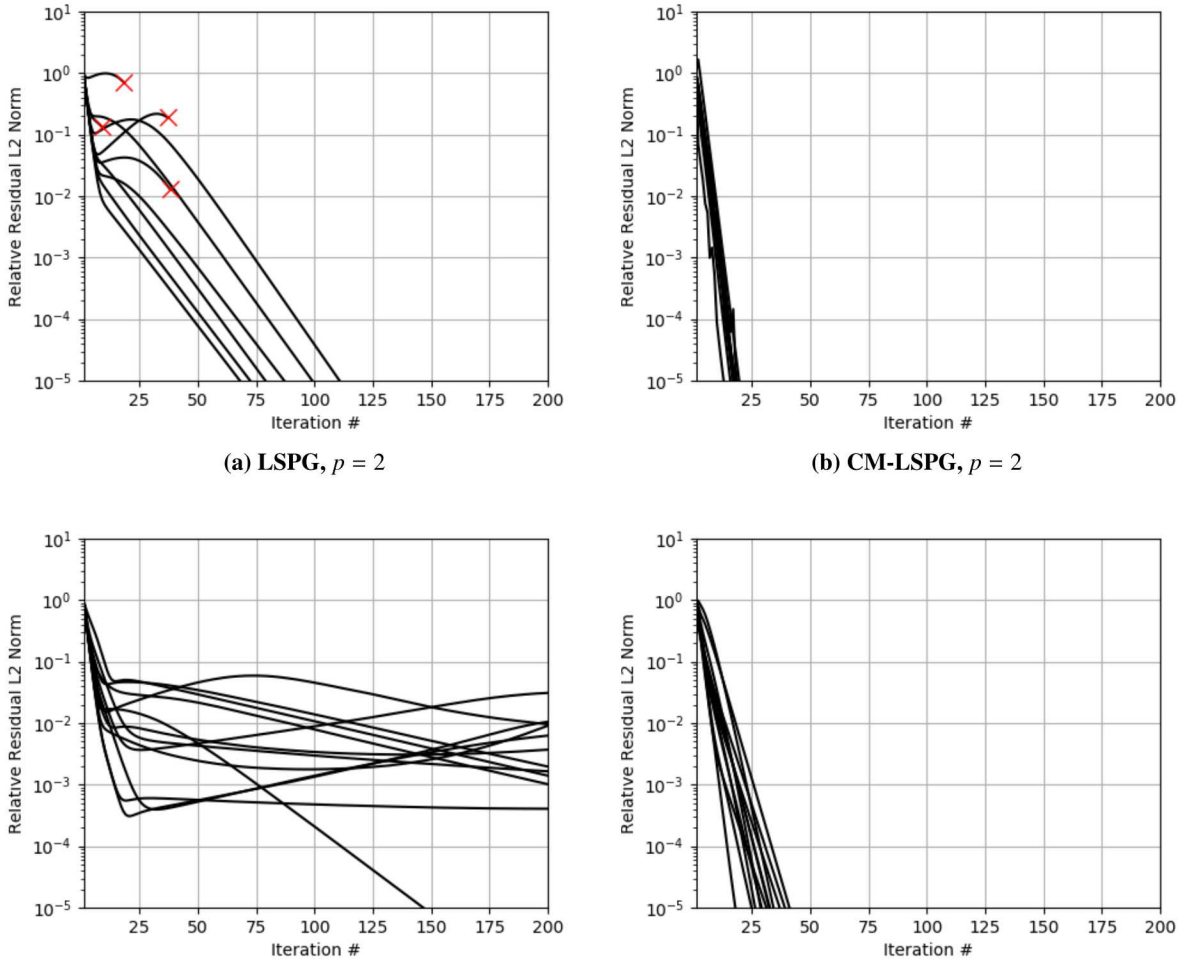

**Fig. 5 Ratio of $\mathcal{E}_{Q_{wall}}$ between the $p = 4$ CM-ROM initial guess and solution. Circle areas are proportional to the error ratio. Larger values indicate cases for which the ROM has a smaller $\mathcal{E}_{Q_{wall}}$ than the initial guess.**

of $\mathcal{E}_{Q_{wall}}$ with a lower ROM dimension. This result is counter-intuitive, but is consistent with the fact that although LSPG guarantees a monotonic reduction in $\mathcal{E}_{\boldsymbol{x}}$ as $p$ is increased, there is no such guarantee for error of derived field or quantities such as integrated wall heat flux.

In addition to being more accurate and robust, CM-LSPG converges in far fewer Gauss-Newton iterations than LSPG, as shown in Figure 6. For ROM dimension $p = 2$, Figure 6a shows that the relative residual norm for LSPG reaches $10^{-5}$ in roughly 60-110 iterations, while Figure 6b shows that C-LSPG converges to the same relative residual norm in only 10-20 iterations. For $p = 4$, the residual norm stops decreasing or increases for some LSPG cases after roughly 20 Gauss-Newton iterations, as shown in Figure 6c. Additionally, the CM-LSPG residuals for $p = 4$ converge at slightly slower than $p = 2$. The degradation of nonlinear convergence rates as the ROM dimension is increased is likely due to the increased stiffness of the LSPG Jacobian for larger values of $p$; similar to the increased stiffness observed for Jacobians of finite element discretizations as the element polynomial basis order is increased.

*2. ROMs with hyper-reduction*

Although the CM-LSPG ROM is accurate and robust, its cost scales with the number of degrees of freedom in the FOM, $N$. Recall from section III that hyper-reduction can break this scaling, allowing for potentially inexpensive ROMs that only require $\boldsymbol{f}$ to be computed for a few cells. In this section, we demonstrate LSPG, C-LSPG, and CM-LSPG with hyper-reduction for a fixed ROM dimension, $p = 4$. We present results for two sample meshes, shown in Figure 7, which we hereafter refer to as sample mesh A and B. Sample mesh A, see Figure 7(a), contains roughly 1.0% of the full



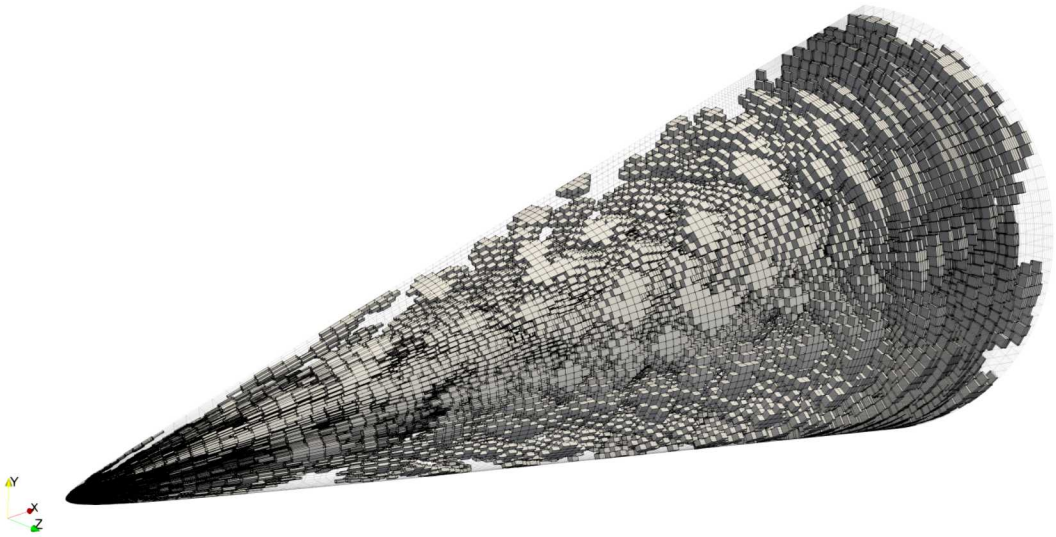

**(a) LSPG, $p = 2$**

**(b) CM-LSPG, $p = 2$**

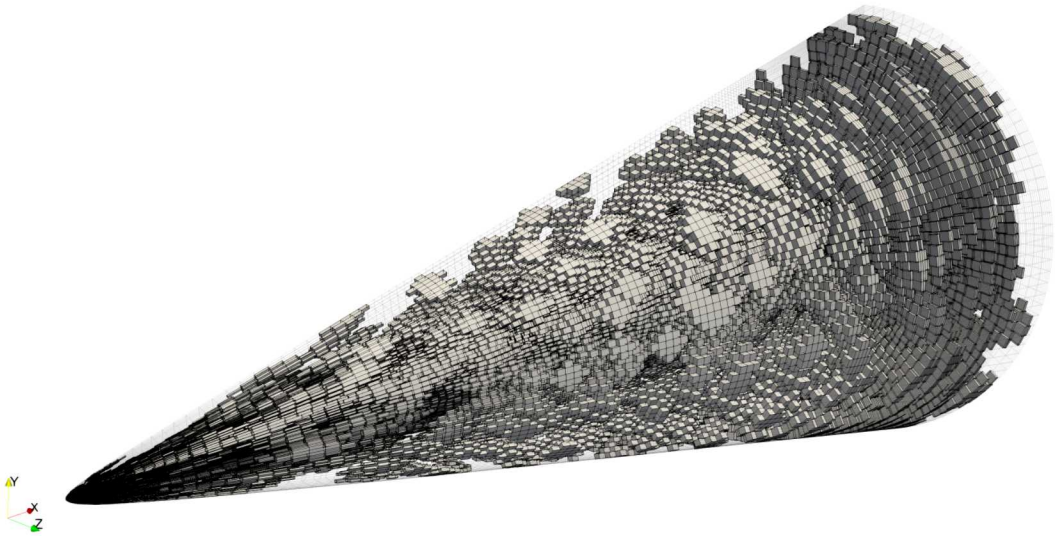
**(c) LSPG, $p = 4$**

**(d) CM-LSPG, $p = 4$**

**Fig. 6 Relative residual L2 norms for LSPG and CM-LSPG for different ROM dimensions $p$. Red X's mark iteration on which cases failed.**




**(a) Sample mesh A. The residual is sampled at 20316 randomly selected cells (1.0% of all cells), requiring 445373 cells (21.9% of all cells) to keep neighbor cells and neighbors of neighbors.**




**(b) Sample mesh B. The residual is sampled at 16253 randomly selected cells (0.8% of all cells), requiring 364468 total cells (17.9% of all cells) to keep neighbor cells and neighbors of neighbors.**

**Fig. 7 HIFiRE-1 mesh with sample mesh cells highlighted. Note that sample mesh B is a subset of sample mesh A.**

**Table 4 Percent relative L2 error, $100\mathcal{E}_{\boldsymbol{x}}$, computed for the state vector for different cases and Sample meshes. Case numbers correspond to the labels in Figure 4. Labels and abbreviations are defined in table 2.**

| | Full Mesh | | | Sample Mesh A | | | Sample Mesh B | | |
|---|---|---|---|---|---|---|---|---|---|
| Case | IG | LSPG | CM-LSPG | IG | LSPG | CM-LSPG | IG | LSPG | CM-LSPG |
| 0 | 3.42 | 2.02 | 0.16 | 3.42 | 4.18 | 0.18 | 3.42 | 5.13 | 0.24 |
| 1 | 2.86 | 1.92 | 0.19 | 2.86 | 0.49 | 0.20 | 2.86 | 0.25 | **0.96** |
| 2 | 4.45 | 7.10 | 0.12 | 4.45 | 9.71 | 0.12 | 4.45 | 9.68 | 0.13 |
| 3 | 8.05 | 0.47 | 0.12 | 8.05 | 0.11 | 0.15 | 8.05 | 0.25 | 0.17 |
| 4 | 18.70 | 0.49 | 0.20 | 18.70 | N/A | 0.28 | 18.70 | N/A | **1.00** |
| 5 | 4.57 | 4.90 | 0.20 | 4.57 | 7.82 | 0.21 | 4.57 | 6.49 | 0.27 |
| 6 | 9.76 | 2.30 | 0.15 | 9.76 | 5.43 | 0.15 | 9.76 | 4.90 | **1.12** |
| 7 | 2.95 | 1.86 | 0.17 | 2.95 | 2.12 | 0.17 | 2.95 | 1.06 | 0.29 |
| 8 | 6.90 | 2.28 | 0.34 | 6.90 | 3.35 | 0.44 | 6.90 | 3.23 | 0.69 |
| 9 | 0.80 | 1.55 | 0.12 | 0.80 | 5.63 | 0.13 | 0.80 | 2.89 | 0.13 |
| 10 | 11.64 | 2.51 | 0.22 | 11.64 | 1.14 | 0.22 | 11.64 | 1.24 | 0.21 |
| 11 | 12.37 | 0.30 | 0.20 | 12.37 | 1.34 | 0.32 | 12.37 | 1.59 | 0.44 |

mesh, while sample mesh B, see Figure 7(b), contains 0.8% of the full mesh cells.

Table 4 shows the state error results. As for the full mesh ROM, C-LSPG and CM-LSPG are more accurate than LSPG, with lower values of $\mathcal{E}_{\boldsymbol{x}}$ in almost all cases. CM-LSPG performs very well: although $\mathcal{E}_{\boldsymbol{x}}$ is mostly larger than that computed for CM-LSPG on the full mesh, it is still roughly 1% or less for both sample meshes. Table 4 also shows that CM-LSPG is more robust than LSPG and C-LSPG, since the clipper functions are active for three cases computed with sample mesh B. Note that the hyper-reduced CM-LSPG is only approximately conservative since we also compute the conservation constraint with the approximate residual $\boldsymbol{P}_r^T \boldsymbol{P}_r \boldsymbol{f}$ computed from the sample mesh.

Table 5 shows the results for the wall heat flux, revealing that the hyper-reduced ROMs have mostly similar values the full mesh results. The ROMs only clearly outperform the initial guess for 5 of 12 test cases, and for most cases, $\mathcal{E}_{Q_{wall}}$ is smaller for CM-LSPG solutions than LSPG solutions. Unlike $\mathcal{E}_{\boldsymbol{x}}$, $\mathcal{E}_{Q_{wall}}$ can be smaller for the sample mesh than the full mesh; see for instance cases 1 and 9 for sample mesh A.

For C-LSPG and CM-LSPG, the $\mathcal{E}_{\boldsymbol{x}}$ shows that the smaller sample mesh (i.e. sample mesh B) is slightly less accurate in all cases except case 10. The impact of sample mesh for LSPG is less clear, likely due to the poor convergence of the residual shown in Figure 8. Figure 8 also shows that the C-LSPG and CM-LSPG residuals converge faster than the corresponding LSPG residuals, as observed for the full mesh. However, the convergence of the residual magnitude for both sample meshes is slower than that on the full mesh in some cases. In one case for sample mesh B, the residual fails to drop by 5 orders of magnitude in 200 iterations. Coarser sample meshes not presented in this paper resulted in C-LSPG and CM-LSPG residual convergence that is slower than the worst cases shown for sample mesh B.

In contrast to C-LSPG and CM-LSPG, the convergence behavior of LSPG tends to improve on smaller sample

**Table 5 Percent relative error, $100\mathcal{E}_{Q_{wall}}$, computed for the integrated wall heat flux for different cases and sample meshes. Labels are defined in table 2.**

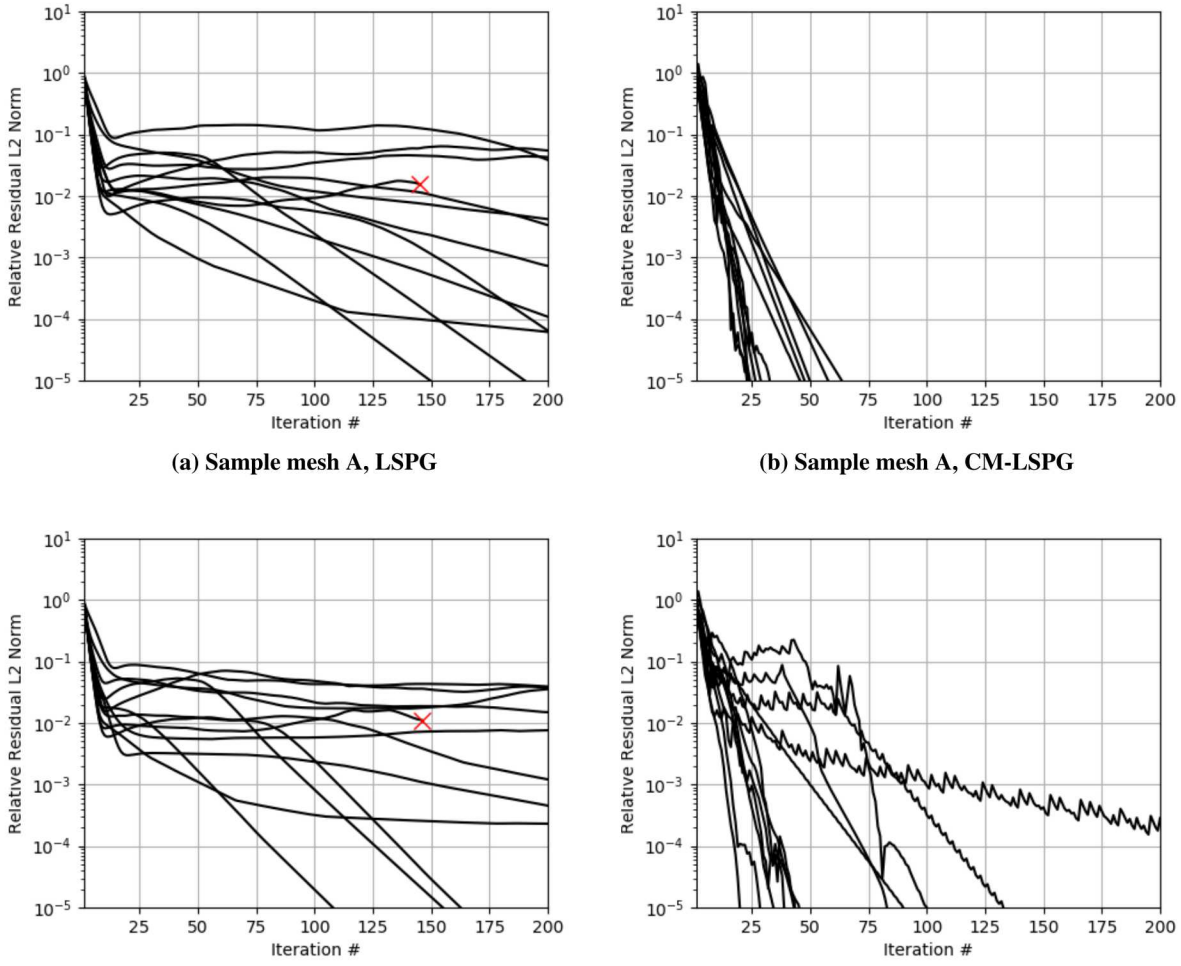
| | Full Mesh | | | Sample Mesh A | | | Sample Mesh B | | |
|---|---|---|---|---|---|---|---|---|---|
| Case | IG | LSPG | CM-LSPG | IG | LSPG | CM-LSPG | IG | LSPG | CM-LSPG |
| 0 | 0.07 | 2.03 | 1.01 | 0.07 | 3.13 | 1.02 | 0.07 | 3.76 | 0.91 |
| 1 | 1.55 | 1.70 | 0.57 | 1.55 | 0.21 | 0.38 | 1.55 | 0.36 | **1.09** |
| 2 | 2.10 | 4.13 | 1.02 | 2.10 | 5.14 | 1.03 | 2.10 | 5.12 | 1.05 |
| 3 | 10.05 | 1.55 | 1.39 | 10.05 | 1.39 | 1.46 | 10.05 | 1.47 | 1.44 |
| 4 | 17.46 | 1.80 | 1.85 | 17.46 | N/A | 1.63 | 17.46 | N/A | **1.10** |
| 5 | 2.30 | 2.33 | 1.23 | 2.30 | 3.52 | 1.36 | 2.30 | 2.98 | 1.42 |
| 6 | 8.24 | 2.96 | 1.24 | 8.24 | 4.47 | 1.29 | 8.24 | 4.25 | **1.61** |
| 7 | 1.79 | 1.81 | 0.55 | 1.79 | 0.73 | 0.45 | 1.79 | 1.20 | 0.58 |
| 8 | 3.75 | 2.45 | 1.28 | 3.75 | 0.29 | 1.29 | 3.75 | 0.36 | 0.97 |
| 9 | 0.05 | 0.63 | 0.56 | 0.05 | 2.63 | 0.52 | 0.05 | 1.37 | 0.53 |
| 10 | 9.35 | 4.44 | 1.87 | 9.35 | 3.16 | 1.93 | 9.35 | 3.25 | 1.96 |
| 11 | 15.56 | 2.08 | 2.24 | 15.56 | 2.37 | 2.36 | 15.56 | 2.43 | 2.55 |

meshes. This can be seem for three cases in Figures 8a and 8c, and was also observed for smaller sample meshes whose results are not plotted for brevity. The convergence acceleration of LSPG on smaller meshes suggests that the convergence issues observed for C-LSPG and CM-LSPG on coarse sample meshes are related to the conservation constraint. In fact, the convergence of the constraint equation solutions also degrades as the sample mesh is coarsened. This degradation in convergence could result from the poor approximation of the constraint with a collocated residual in (9). If this is the case, then employing a hyper-reduction scheme that approximates the full residual more accurately such as GNAT may improve the convergence of C-LSPG and CM-LSPG on sample meshes.

*3. ROM performance*

To estimate the performance of LSPG, C-LSPG, and CM-LSPG, we compared the CPU times of the $p = 4$ dimension ROMs with the FOMs for all 12 test cases. Table 6 shows that the hyper-reduced ROMs are between 200 and 1,000 faster than the FOM at the test cases. As in [11], these speed-up estimates include the time needed to compute $\boldsymbol{q}_{wall}$ as well as other wall quantities, e.g. pressure and friction coefficients, on a second post-processing mesh that only contained the two layers of cells nearest to the wall needed to compute wall normal gradients. The speed-ups for C-LSPG and CM-LSPG are very similar, highlighting the negligible cost of the clipper function implementation used in this study. The size of this speed-up means that hundreds of ROMs could be run with the same computational resources required for one FOM, allowing many-query analyses to be applied to steady hypersonic CFD cases like the one considered in this paper.

Table 6 shows that the full mesh ROM is hundreds of times faster than the FOM. This is partially because the initial guess supplied to the ROM is more accurate than the uniform flow initial condition used by the FOM, but also related to




**(a) Sample mesh A, LSPG**

**(b) Sample mesh A, CM-LSPG**

**(c) Sample mesh B, LSPG**

**(d) Sample mesh B, CM-LSPG**

**Fig. 8 Relative residual L2 norms for hyper-reduced LSPG and CM-LSPG computed with ROM dimension $p = 4$ on both sample meshes. Red X's mark iteration on which cases failed.**

the much smaller dimension of the LSPG minimization problem, $p << N$. The low relative cost of the full mesh ROM means the GNAT hyper-reduction technique [11], which requires residual snapshots taken from full mesh ROMs, may be feasible for steady flow problems.

## V. Conclusion

This work demonstrated the successful application of LSPG ROMs to steady-state flows around a hypersonic flight vehicle. We showed how to obtain accurate approximations of conserved quantity fields (the state vector) and derived quantity fields, such as wall heat flux, at a fraction of the cost of the full order model. The methods thus seem promising towards enabling many-query analyses for large steady state hypersonic flow simulations.

The results of section IV indicate that conservative manifold LSPG (CM-LSPG) is accurate and fairly robust when

**Table 6 Speed-up of the ROM over full order model for LSPG, C-LSPG, and CM-LSPG. Case numbers correspond to the labels in Figure 4. Labels and abbreviations are defined in table 2. Entries are colored by their speed-up value $\xi$; red for speed-ups $\xi < 100$, orange for errors $100 < \xi \leq 200$, yellow for errors $200 < \xi \leq 500$, and green for $\xi > 500$. The ROMs and FOMs were both run on 2.6 GHz Intel Sandy Bridge processors. The FOM cases were run with 128 cores, while the ROM cases were run on 16 cores, corresponding to a single cluster node. Wall time speed ups can be obtained by dividing the results in this plot by 8. Note that ROM times include a postprocessing step in which the flow field was computed on a mesh containing the near-wall cells needed to compute wall quantities like heat flux, as in [10].**

| | Full Mesh | | | Sample Mesh A | | | Sample Mesh B | | |
|---|---|---|---|---|---|---|---|---|---|
| Case | LSPG | C-LSPG | CM-LSPG | LSPG | C-LSPG | CM-LSPG | LSPG | C-LSPG | CM-LSPG |
| | 53 | 161 | 164 | 155 | 716 | 607 | 186 | 366 | 322 |
| 1 | 52 | 171 | 173 | 146 | 474 | 474 | 176 | N/A | 324 |
| 2 | 51 | 198 | 203 | 146 | 715 | 716 | 176 | 771 | 771 |
| 3 | 124 | 391 | 396 | 353 | 1071 | 1117 | 569 | 1041 | 1056 |
| 4 | 52 | 189 | 193 | N/A | 499 | 498 | N/A | N/A | 704 |
| 5 | 50 | 189 | 192 | 141 | 701 | 683 | 173 | 567 | 562 |
| 6 | 51 | 183 | 186 | 149 | 387 | 385 | 180 | N/A | 379 |
| 7 | 52 | 162 | 164 | 148 | 416 | 409 | 177 | 180 | 176 |
| 8 | 64 | 228 | 231 | 193 | 815 | 804 | 280 | 768 | 760 |
| 9 | 51 | 178 | 176 | 148 | 462 | 460 | 179 | 255 | 257 |
| 10 | 51 | 238 | 244 | 144 | 731 | 730 | 177 | 939 | 940 |
| 11 | 73 | 272 | 278 | 211 | 1008 | 1008 | 308 | 949 | 962 |

trained with sufficient snapshot data over a parameter range. The 2-parameter CM-LSPG ROM state errors are 0.1% and heat flux errors were at most 1-2%. Hyper-reduction via collocation with random cell sampling maintains a high level of accuracy for the state vector and derived fields like integrated wall heat flux while increasing performance substantially over the full mesh ROM. This is particularly true for CM-LSPG, despite the crude approximation of the conservation constraint by collocation.

One important observation is that the utility of the ROM depends on the region of the parameter space it is used in. The ROM solution is substantially more accurate than the initial guess for cases on the edge of the training set parameter space, where state and heat flux errors are around 5-10 times smaller than the initial guess, which was as high as 10-20% for some cases. However, the initial guess can be sufficiently accurate for regions of parameter space that are well inside the basis training set. The initial guess is computed with a low-cost surrogate model: inverse-distance interpolation over POD modes. Therefore, the most efficient way to compute many approximate solutions as part of a many-query analysis will likely involve using ROMs in some parts of parameter space, and surrogate models in other regions.

Finally, we found that using a naive Gauss-Newton implementation with a linear basis can result in the solver finding non-physical solutions with regions of negative temperature. These non-physical states can be avoided by using a nonlinear basis which does not allow for negative densities or temperatures. The increased robustness achieved by avoiding non-physical states will enable ROMs to be trained with fewer snapshots and/or trained over wider parameter

ranges.

There are a large number of future research directions following up the results presented in this paper. First, to apply CM-LSPG projection to models of flight vehicles that operate in flight regimes where non-equilibrium effects are important, appropriate clipping functions need to be derived for non-perfect gas models. Second, although collocation with random cell selection works well, there are likely better hyper-reduction strategies that can improve upon the accuracy and/or performance. The speed-ups obtained for CM-LSPG on the full mesh suggest that GNAT hyper-reduction may be feasible for steady state hypersonic flow simulations. Third, the poor nonlinear convergence rates observed for some LSPG and CM-LSPG cases suggests that more work is needed on the LSPG and CM-LSPG solvers. In particular, better preconditioning strategies may help improve nonlinear convergence rates. Additionally, finding a more accurate initial guess than the inverse-distance interpolation could substantially improve nonlinear convergence rates.

With further improvements in robustness, accuracy and performance, ROMs will enable large-scale UQ studies and other many-query analyses of hypersonic CFD models, enabling further advances in our understanding of CFD model uncertainties and vehicle design.

# Appendix

## A. Governing Equations

We consider the three–dimensional compressible Navier–Stokes equations with a turbulence model, which corresponds to Eqn. (1) with $d = 3$, and $n_u = 5, 6$ or 7 depending on the choice of turbulence model[‡]. The conserved quantities are written in vector form as

$$\boldsymbol{U} \equiv \begin{pmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \\ u_5 \\ u_6 \\ \vdots \\ u_{n_u} \end{pmatrix} = \begin{pmatrix} \rho \\ \rho v_1 \\ \rho v_2 \\ \rho v_3 \\ \rho E \\ \rho \phi_1 \\ \vdots \\ \rho \phi_{n_t} \end{pmatrix} \tag{26}$$

where $\rho$ is density of the fluid, $\rho v_j$ is the fluid density times the fluid velocity $v_j$, $\rho E$ is the fluid density times the

[‡]Note that additional conservation equations are required for non-perfect gases, and non-equilibrium chemistry, but these will not be considered in this paper.

total energy per unit mass $E$, and $\phi_t$ is a set of scalars that belong to turbulent transport equations and $n_t$ is the number of turbulent transport equations. For the case of $n_t = 0$, no turbulence equation is used and the set of equations are the 'laminar' perfect gas equations. For the case of $n_t \geq 1$, each turbulent transport variable will have an associated inviscid flux, viscous flux and source term entry (e.g. $\phi_1 = \tilde{\nu}$ in the case of the one equation Spalart-Allmaras turbulence model). The total energy per unit mass is the sum of the fluid's internal energy $e$ and kinetic energy and can be written as

$$E = e + \frac{1}{2}(v_j v_j). \tag{27}$$

The fluxes $\boldsymbol{g}_i$ can be decomposed into inviscid, $\boldsymbol{F}_i$, and viscous, $\boldsymbol{G}_i$, flux vectors as

$$\begin{pmatrix} \boldsymbol{g}_1 \\ \vdots \\ \boldsymbol{g}_{n_u} \end{pmatrix} = \boldsymbol{F}_i(\boldsymbol{U}) - \boldsymbol{G}_i(\boldsymbol{U}). \tag{28}$$

The inviscid flux vector $\boldsymbol{F}_i$ is defined as

$$\boldsymbol{F}_i(\boldsymbol{U}) = \begin{pmatrix} \rho v_i \\ \rho v_i v_j + P\delta_{ij} \\ \rho E v_i + P v_i \\ F^{\text{turb}}_{i,1} \\ \vdots \\ F^{\text{turb}}_{i,n_t} \end{pmatrix}, \tag{29}$$

where $P$ is the pressure of the fluid and $F^{\text{turb}}_{i,t}$ denotes the turbulent inviscid flux in the $i$-th Cartesian direction associated with the $t$-th turbulent transport variable. The viscous flux vector $\boldsymbol{G}_i$ is written by

$$\boldsymbol{G}_i(\boldsymbol{U}) = \begin{pmatrix} 0 \\ \tau_{ij} \\ \tau_{ij} v_j - q_i \\ G^{\text{turb}}_{i,1} \\ \vdots \\ G^{\text{turb}}_{i,n_t} \end{pmatrix}, \tag{30}$$

where $\tau_{ij}$ and $q_i$ are the viscous stress tensor and the heat flux vector, respectively, and represent diffusive effects of the fluid. Similar to the turbulent inviscid fluxes, $G_{i,t}^{\text{turb}}$ are the viscous fluxes associated with the $t$-th turbulent transport variable. In addition to the advection transport mechanism associated with the motion of the fluid, the fluid has the ability to transport momentum and energy via a diffusion process. In the absence of any diffusion, the viscous Navier–Stokes equations reduce to the *inviscid Euler equations* which account solely for advection. Since viscous effects are of primary concern for most practical aerodynamic problems, the Euler equations will not be further discussed.

The viscous stress tensor $\tau_{ij}$ requires a constitutive equation which relates the viscosity and spatial derivatives of the velocity to the stresses. For a Newtonian fluid (i.e. one which has a linear stress/strain relationship) the deviatoric stress tensor is often written as

$$\tau_{ij} = \mu \left( \frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i} \right) + \lambda \delta_{ij} \left( \frac{\partial v_k}{\partial x_k} \right), \tag{31}$$

where $\mu$ is the viscosity and $\lambda$ is the bulk viscosity of the fluid. For a Newtonian fluid the bulk viscosity is often expressed as $\lambda = -2\mu/3$.

The heat flux vector $q_i$ is a measure of the thermal energy flow and is typically written using Fourier's law

$$q_i = -\kappa \frac{\partial T}{\partial x_i}, \tag{32}$$

where $\kappa$ is the gas thermal conductivity and $T$ is the gas temperature.

Lastly the source vector $\boldsymbol{S}$ is written as

$$\boldsymbol{S}(\boldsymbol{U}) \equiv \begin{pmatrix} s_1 \\ \vdots \\ s_{n_u} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ S_1 \\ \vdots \\ S_{n_t} \end{pmatrix}, \tag{33}$$

where $S_t$ is the source term contribution from each turbulent transport equation.

**Equation of State** For a calorically perfect gas, an equation of state is needed to relate two independent state variables to the third. Thus, the perfect gas equation of state is usually written as

$$P = \rho RT, \tag{34}$$

where $R$ is a constant specific to the type of gas (for air $R = 287.1\ J/kg/K$). The calorically perfect gas assumption has the following requirements: (a) the gas is in thermal equilibrium, (b) the gas is not chemically reacting, (c) the internal energy and enthalpy are dependent only on temperature, and (d) the specific heats ($c_\mathrm{v}$ and $c_\mathrm{p}$) are constant.

In accordance with these assumptions, the internal energy and enthalpy are computed by the equations

$$e = c_\mathrm{v} T \quad , \quad h = c_\mathrm{p} T, \tag{35}$$

and the specific heats are written as

$$c_\mathrm{v} = \frac{R}{\gamma - 1} \quad , \quad c_\mathrm{p} = \frac{\gamma R}{\gamma - 1}, \tag{36}$$

where $\gamma$ is the ratio of specific heats (for air $\gamma = 1.4$) and is expressed as

$$\gamma = \frac{c_\mathrm{p}}{c_\mathrm{v}}. \tag{37}$$

An alternative but equivalent form of the perfect gas equation of state can be obtained by writing the temperature as $T = e(\gamma - 1)/R$. Inserting this temperature expression into Eq. (34) we obtain the following form of the ideal gas equation

$$P = (\gamma - 1)\rho e. \tag{38}$$

The Mach number is defined as the fluid speed divided by the speed of sound $C$

$$M = \frac{\sqrt{v_j v_j}}{C}, \tag{39}$$

and the speed of sound is computed by

$$C = \sqrt{\gamma \frac{P}{\rho}}. \tag{40}$$

**Transport Properties** The viscous stress tensor $\tau_{ij}$ and heat flux vector $q_i$ rely on transport coefficients that determine the rate of the diffusion process. The viscosity coefficient for a gas is a macroscopic approximation of momentum transport within the flow as a result of molecular diffusion. Several models for the viscosity of a gas exist, with the most

common probably being Sutherland's law. The Sutherland formula is written in two coefficient form as

$$\mu = \mu_{\text{ref}} \frac{T^{3/2}}{T + T_{\text{ref}}}, \tag{41}$$

For air at temperatures below roughly $1000K$ and pressures below around $1 \times 10^6\ N/m^2$, valid reference values are $\mu_{\text{ref}} = 1.458 \times 10^{-6}\ kg/m \cdot s \cdot K^{1/2}$ and $T_{\text{ref}} = 110.4K$. The Sutherland formula may also be written in a three coefficient form as

$$\mu = \mu_{\text{ref}} \left(\frac{T}{T_{\text{ref}}}\right)^{3/2} \frac{T_{\text{ref}} + S}{T + S}, \tag{42}$$

where $\mu_{\text{ref}} = 1.716 \times 10^{-5}\ kg/m \cdot s$, $T_{\text{ref}} = 273.11K$ and $S = 110.56K$.

The coefficient of thermal conductivity needed for the heat flux computation is a measure of the energy transport resulting from molecular collisions. The thermal conductivity of a gas is often modeled as a relation of the Prandtl number and viscosity by the equation

$$\kappa = \frac{c_{\text{p}}\mu}{Pr}, \tag{43}$$

where $c_{\text{p}}$ is the specific heat of the gas at constant pressure, $Pr$ is the Prandtl number and $\mu$ is the viscosity of the fluid. The Prandtl number is the ratio of the viscous diffusion rate to the thermal diffusion rate and for laminar flow of air at moderate temperatures the Prandtl number is assumed to be constant and equal to approximately 0.71.

**Boundary Conditions** The system of equations presented in (1) are completed by a set of boundary conditions

$$\boldsymbol{b}(\boldsymbol{U}) = \bar{\boldsymbol{b}}(\boldsymbol{U}, \vec{x}; \boldsymbol{\mu}) \quad \text{on} \quad \Gamma_f, \tag{44}$$

which prescribe the values $\bar{\boldsymbol{b}}$ of a general nonlinear boundary condition $\boldsymbol{b}$ through time. Flux boundary conditions may be imposed such that

$$\boldsymbol{F}_i(\boldsymbol{U}) = \bar{\boldsymbol{F}}_i(\boldsymbol{U}, \vec{x}; \boldsymbol{\mu}) \quad \text{on} \quad \Gamma_{\bar{f}}, \tag{45}$$

and

$$\boldsymbol{G}_i(\boldsymbol{U}) = \bar{\boldsymbol{G}}_i(\boldsymbol{U}, \vec{x}; \boldsymbol{\mu}) \quad \text{on} \quad \Gamma_{\bar{f}}. \tag{46}$$

Additionally, the conservative variables $\boldsymbol{U}$ must be specified at each point $\boldsymbol{x}$ as initial conditions at $t = 0$

$$\boldsymbol{U}(\vec{x}, t = 0; \boldsymbol{\mu}) = \boldsymbol{U}_0(\vec{x}; \boldsymbol{\mu}) \quad \text{in} \quad \Omega_f. \tag{47}$$

### B. Proper Orthogonal Decomposition

Table 7 shows the cumulative statistical energy for different basis sizes for the training data shown in Figure 4. Table 8 shows the maximum absolute value of each conserved quantity over the training snapshots.

**Table 7 Cumulative energy for case 1 POD Basis $\bar{\Phi}$.**

| Mode | Cumulative Statistical Energy |
|---|---|
| 1 | 0.92208 |
| 2 | 0.99191 |
| 3 | 0.99890 |
| 4 | 0.99972 |
| 5 | 0.99988 |
| 6 | 0.99993 |
| 7 | 0.99995 |
| 8 | 0.99997 |

**Table 8 Case 1 conserved variable scales in $D_{max}$.**

| Conserved Variable | Maximum Absolute Value | Units |
|---|---|---|
| $\rho$ | 0.2695 | $kg/m^3$ |
| $\rho v_1$ | 97.52 | $kg/(m^2 s)$ |
| $\rho v_2$ | 103.6 | $kg/(m^2 s)$ |
| $\rho v_3$ | 101.7 | $kg/(m^2 s)$ |
| $\rho E$ | 164900 | $kgJ/(m^3)$ |
| $\rho \phi_1$ | 0.006325 | $kg^2/(m^2 s)$ |

## Funding Sources

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.

## Acknowledgments

## References

[1] Lamorte, N., Friedmann, P. P., Glaz, B., Culler, A. J., Crowell, A. R., and McNamara, J. J., "Uncertainty Propagation in Hypersonic Aerothermoelastic Analysis," *Journal of Aircraft*, Vol. 51, No. 1, 2014, pp. 192–203. https://doi.org/10.2514/1.C032233, URL https://doi.org/10.2514/1.C032233.

[2] West, T. K., Hosder, S., and Johnston, C. O., "Multistep Uncertainty Quantification Approach Applied to Hypersonic Reentry Flows," *Journal of Spacecraft and Rockets*, Vol. 51, No. 1, 2014, pp. 296–310. https://doi.org/10.2514/1.A32592, URL https://doi.org/10.2514/1.A32592.

[3] West, T. K., and Hosder, S., "Uncertainty Quantification of Hypersonic Reentry Flows with Sparse Sampling and Stochastic

Expansions," *Journal of Spacecraft and Rockets*, Vol. 52, No. 1, 2015, pp. 120–133. https://doi.org/10.2514/1.A32947, URL https://doi.org/10.2514/1.A32947.

[4] Dutta, S., Braun, R. D., and Karlgaard, C. D., "Uncertainty Quantification for Mars Entry, Descent, and Landing Reconstruction Using Adaptive Filtering," *Journal of Spacecraft and Rockets*, Vol. 51, No. 3, 2014, pp. 967–977. https://doi.org/10.2514/1.A32716, URL https://doi.org/10.2514/1.A32716.

[5] Dalle, D. J., Frendreis, S. G. V., Driscoll, J. F., and Cesnik, C. E. S., "Hypersonic Vehicle Flight Dynamics with Coupled Aerodynamics and Reduced-order Propulsive Models," *AIAA Atmospheric Flight Mechanics Conference*, AIAA 2010-7930, Vol. 1, AIAA, Toronto, Canada, 2010. https://doi.org/10.2514/6.2010-7930.

[6] Falkiewicz, N. J., Cesnik, C. E. S., Crowell, A. R., and J, M. J., "Proper Orthogonal Decomposition for Reduced-Order Thermal Solution in Hypersonic Aerothermoelastic Simulations," *AIAA Journal*, Vol. 49, No. 8, 2011, pp. 1625, 1647. https://doi.org/10.2514/1.J050802.

[7] Crowell, A. R., and McNamara, J. J., "Model Reduction of Computational Aerothermodynamics for Hypersonic Aerothermoelasticity," *AIAA Journal*, Vol. 50, No. 1, 2012, pp. 74, 85. https://doi.org/10.2514/1.J051094.

[8] 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. https://doi.org/10.2514/1.J050701.

[9] 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. https://doi.org/10.2514/1.J055499.

[10] 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.

[11] 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.

[12] 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.

[13] 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.

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

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

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

[17] 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.

[18] 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.

[19] Huang, C., Duraisamy, K., and Merkle, C., *Investigations and Improvement of Robustness of Reduced-Order Models of Reacting Flow*, AIAA, San Diego, CA, 2019, chapter and pages. https://doi.org/10.2514/6.2019-2012, URL https://arc.aiaa.org/doi/abs/10.2514/6.2019-2012.

[20] Lee, K., and Carlberg, K., "Model reduction of dynamical systems on nonlinear manifolds using deep convolutional autoencoders," , 2018. URL http://arxiv.org/abs/1812.08373, submitted to the Journal of Computational Physics, Dec. 2018.

[21] 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. https://doi.org/10.2514/6.2017-4407, URL https://arc.aiaa.org/doi/abs/10.2514/6.2017-4407.

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

[23] Wadhams, T. P., Mundy, E., MacLean, M. G., and Holden, M. S., "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. https://doi.org/10.2514/1.38338, URL https://doi.org/10.2514/1.38338.

[24] Brunini, V., Carnes, B., Dinzl, D., Freno, B., Hammond, S., Hoemmen, M., Howard, M., Keedy, R., Kieweg, S., Kim, K., Mussoni, E., Ray, J., Sjaardema, G., Watkins, J., Weirs, G., and Winokur, J., "ASC ATDM FY19 Level 2 Milestone Report: SPARC Credibility, Performance and Scaling," SAND Report SAND2019-10842, Sandia National Laboratories, 2019.

[25] Washabaugh, K. M., "Fast Fidelity for Better Design: A Scalable Model Order Reduction Framework for Steady Aerodynamic Design Applications," PhD dissertation, Stanford University, Department of Aeronautics and Astronautics, Aug. 2016.