SAND2019-14636C

Model reduction for hypersonic aerodynamics via conservative LSPG projection and hyper-reduction

Patrick J. Blonigan* and Kevin T. Carlberg†
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

High-speed aerospace engineering applications rely heavily on computational fluid dynamics (CFD) models for design and analysis due to the expense and difficulty of flight tests and experiments. This reliance on CFD models necessitates performing accurate and reliable uncertainty quantification (UQ) of the CFD models. However, it is very computationally expensive to run CFD for hypersonic flows due to the fine grid resolution required to capture the strong shocks and large gradients that are typically present. 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 [1, 2]. This allows computationally inexpensive LSPG ROM simulations to replace the full-order model simulations in UQ studies, which makes this many-query task tractable, even for large-scale CFD models.

This work presents the first application of LSPG to a hypersonic CFD application. In particular, we present results for LSPG ROMs of the HIFiRE-1 in a three-dimensional, turbulent Mach 7.1 flow, showcasing the ability of the ROM to significantly reduce computational costs while maintaining high levels of accuracy in computed quantities of interest.

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 [3–6].

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 in the model. 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 that apply surrogate and low-fidelity approaches to hypersonic aerodynamics models. Many of these proposed aerodynamic surrogates are low-fidelity models arising from simplified physics,

*Extreme-Scale Data Science and Analytics, MS 9159, AIAA Member
†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

including two-dimensional oblique shock relations [7] and piston-theory aerodynamics [8]. One study by Crowell and McNamara 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 [9]. It should be noted that while Galerkin-projection ROMs have been applied to both linear [10] and nonlinear [11] finite-element heat-transfer models of hypersonic vehicles, they have not yet been applied to hypersonic CFD to our knowledge.

The application of 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 executing a projection process directly on the equations governing the high-fidelity model. We focus on the least-squares Petrov-Galerkin (LSPG) projection [12], due to its observed accuracy and stability on large-scale problems in CFD [1, 2], and its flexible optimization-based formulation that readily admits integration of constraints that enforce conservation laws over subdomains [13].

This paper presents the application of LSPG to hypersonic CFD simulations. To our knowledge, this is the first application of projection-based ROMs to hypersonic CFD simulations. 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. Two test cases are considered; section IV.B presents results for a two-parameter ROM and section IV.C presents a single parameter ROM trained over a wider range of parameters with less data than the first case. Finally, section V offers some conclusions and directions for future work.

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

A. Physical conservation laws

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 in 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, 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; 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 ith conserved variable (per unit volume), the flux associated with the ith conserved variable (per unit area per unit time), and the source associated with the ith conserved variable (per unit volume per unit time). Finally, $\boldsymbol{n}\in\mathbb{R}^d$ denotes the outward unit normal to ω. 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.

B. 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 Ω 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 ith control volume by $\Gamma_i := \partial\Omega_i$. The ith 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

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

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

where $\boldsymbol{n}_j \in \mathbb{R}^d$ denotes the unit normal to control volume Ω_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 ith 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) characterizing an initial value problem, which we consider to be our full-order model (FOM). We thus refer to Eq. (4) as the FOM ODE.

In the case of computing a steady-state solution, we assume that the velocity exhibits no time dependence and set $\dot{\boldsymbol{x}} = 0$ such that the FOM ODE (4) becomes simply

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

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

C. Hypersonic aerodynamics

In this paper, we consider high-Mach external aerodynamics. The enthalpy of the flows we consider are not sufficiently high to drive dissociation of the gas we consider (air), so solving the perfect gas, compressible Navier–Stokes equations is the appropriate choice. The governing equations, equation of state, transport properties and boundary conditions are presented below.

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

$$\boldsymbol{U} \equiv \begin{pmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \\ u_5 \\ u_6 \\ \vdots \\ u_{n_u} \end{pmatrix} = \begin{pmatrix} \rho \\ \rho v_1 \\ \rho v_2 \\ \rho v_3 \\ \rho E \\ \rho \phi_1 \\ \vdots \\ \rho \phi_{n_t} \end{pmatrix} \tag{6}$$

where 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 belong to turbulent transport equations and n_t is the number

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

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

$$E = e + \frac{1}{2}(v_j v_j). \tag{7}$$

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

$$\begin{pmatrix} \boldsymbol{g}_1 \\ \vdots \\ \boldsymbol{g}_{n_u} \end{pmatrix} = \boldsymbol{F}_i(\boldsymbol{U}) - \boldsymbol{G}_i(\boldsymbol{U}). \tag{8}$$

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

$$\boldsymbol{F}_i(\boldsymbol{U}) = \begin{pmatrix} \rho v_i \\ \rho v_i v_j + P\delta_{ij} \\ \rho E v_i + P v_i \\ F^{\text{turb}}_{i,1} \\ \vdots \\ F^{\text{turb}}_{i,n_t} \end{pmatrix}, \tag{9}$$

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

$$\boldsymbol{G}_i(\boldsymbol{U}) = \begin{pmatrix} 0 \\ \tau_{ij} \\ \tau_{ij} v_j - q_i \\ G^{\text{turb}}_{i,1} \\ \vdots \\ G^{\text{turb}}_{i,n_t} \end{pmatrix}, \tag{10}$$

where τ_{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, $G^{\text{turb}}_{i,t}$ 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 τ_{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

$$\tau_{ij} = \mu\left(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i}\right) + \lambda\delta_{ij}\left(\frac{\partial v_k}{\partial x_k}\right), \tag{11}$$

where μ is the viscosity and λ 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

$$q_i = -\kappa\,\frac{\partial T}{\partial x_i}, \tag{12}$$

where κ is the gas thermal conductivity and T is the gas temperature.

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

$$\boldsymbol{S}(\boldsymbol{U}) \equiv \begin{pmatrix} s_1 \\ \vdots \\ s_{n_u} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ S_1 \\ \vdots \\ S_{n_t} \end{pmatrix}, \tag{13}$$

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

$$P = \rho R T, \tag{14}$$

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_v and c_p) are constant.

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

$$e = c_\mathrm{v} T \quad , \quad h = c_\mathrm{p} T, \tag{15}$$

and the specific heats are written as

$$c_\mathrm{v} = \frac{R}{\gamma - 1} \quad , \quad c_\mathrm{p} = \frac{\gamma R}{\gamma - 1}, \tag{16}$$

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

$$\gamma = \frac{c_\mathrm{p}}{c_\mathrm{v}}. \tag{17}$$

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. (14) we obtain the following form of the ideal gas equation

$$P = (\gamma - 1)\rho e. \tag{18}$$

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

$$M = \frac{\sqrt{v_j v_j}}{C}, \tag{19}$$

and the speed of sound is computed by

$$C = \sqrt{\gamma \frac{P}{\rho}}. \tag{20}$$

Transport Properties The viscous stress tensor τ_{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

$$\mu = \mu_\mathrm{ref} \frac{T^{3/2}}{T + T_\mathrm{ref}}, \tag{21}$$

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

$$\mu = \mu_{\text{ref}} \left(\frac{T}{T_{\text{ref}}}\right)^{3/2} \frac{T_{\text{ref}} + S}{T + S}, \tag{22}$$

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

$$\kappa = \frac{c_{\text{p}}\mu}{Pr}, \tag{23}$$

where c_{p} is the specific heat of the gas at constant pressure, Pr is the Prandtl number and μ 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

$$\boldsymbol{b}(\boldsymbol{U}) = \bar{\boldsymbol{b}}(\boldsymbol{U}, \vec{x}; \boldsymbol{\mu}) \quad \text{on} \quad \Gamma_f, \tag{24}$$

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

$$\boldsymbol{F}_i(\boldsymbol{U}) = \bar{\boldsymbol{F}}_i(\boldsymbol{U}, \vec{x}; \boldsymbol{\mu}) \quad \text{on} \quad \Gamma_{\bar{f}}, \tag{25}$$

and

$$\boldsymbol{G}_i(\boldsymbol{U}) = \bar{\boldsymbol{G}}_i(\boldsymbol{U}, \vec{x}; \boldsymbol{\mu}) \quad \text{on} \quad \Gamma_{\bar{f}}. \tag{26}$$

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

$$\boldsymbol{U}(\vec{x}, t = 0; \boldsymbol{\mu}) = \boldsymbol{U}_0(\vec{x}; \boldsymbol{\mu}) \quad \text{in} \quad \Omega_f. \tag{27}$$

D. 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 for them. There are many sources of uncertainty in 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. The large number of uncertain parameters n_μ for hypersonic aerodynamics models make it necessary to evaluate the model many times for many different sets of parameters $\boldsymbol{\mu}$ to obtain reasonable estimates of uncertainty from a UQ approach.

In addition to the large number of uncertain parameters, 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

$$\tilde{\boldsymbol{x}}(t; \boldsymbol{\mu}) = \boldsymbol{x}^0(\boldsymbol{\mu}) + \boldsymbol{\Phi}\hat{\boldsymbol{x}}(t; \boldsymbol{\mu}), \tag{28}$$

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 ℓ^2-norm, i.e.,

$$\hat{\boldsymbol{x}} = \underset{\hat{\boldsymbol{z}}\in\mathbb{R}^p}{\arg\min} \|\boldsymbol{A}\boldsymbol{f}(\boldsymbol{x}^0(\boldsymbol{\mu}) + \boldsymbol{\Phi}\hat{\boldsymbol{z}}; \boldsymbol{\mu})\|_2. \tag{29}$$

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 [1].

B. Conservative LSPG projection

As proposed in Ref. [13], we now modify the LSPG ROM for steady simulations (Eq. (29)) 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

$$\begin{aligned} &\underset{\hat{\boldsymbol{z}}\in\mathbb{R}^p}{\text{minimize}} \ \|\boldsymbol{A}\boldsymbol{f}(\boldsymbol{x}^0(\boldsymbol{\mu}) + \boldsymbol{\Phi}\hat{\boldsymbol{z}}; \boldsymbol{\mu})\|_2 \\ &\text{subject to } \bar{\boldsymbol{C}}\boldsymbol{f}(\boldsymbol{x}; \boldsymbol{\mu}) = \boldsymbol{0}. \end{aligned} \tag{30}$$

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

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

$$\begin{aligned} &\underset{\hat{\boldsymbol{z}}\in\mathbb{R}^p}{\text{minimize}} \ \|\boldsymbol{A}\boldsymbol{f}(\boldsymbol{x}^0(\boldsymbol{\mu}) + \boldsymbol{\Phi}\hat{\boldsymbol{z}}; \boldsymbol{\mu})\|_2 \\ &\text{subject to } \bar{\boldsymbol{C}}\tilde{\tilde{\boldsymbol{f}}}(\boldsymbol{x}; \boldsymbol{\mu}) = \boldsymbol{0}. \end{aligned} \tag{31}$$

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 ith 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 ith 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

$$\int_{\bar{\Gamma}_j} \boldsymbol{g}_i(\vec{x}, t; \boldsymbol{\mu}) \cdot \bar{\boldsymbol{n}}_j(\vec{x}) \, \mathrm{d}\vec{s}(\vec{x}) = \int_{\bar{\Omega}_j} s_i(\vec{x}, t; \boldsymbol{\mu}) \, \mathrm{d}\vec{x}, \quad i \in \mathbb{N}(n_u), \ j \in \mathbb{N}(N_{\bar{\Omega}}), \tag{32}$$

where $\bar{\boldsymbol{n}}_j : \Gamma_j \to \mathbb{R}^d$ 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 (32). 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

$$\bar{x}_{\bar{I}(i,j)}(\boldsymbol{x}; \boldsymbol{\mu}) = \frac{1}{|\bar{\Omega}_j|}\int_{\bar{\Omega}_j} u_i(\vec{x}; \boldsymbol{\mu}) \, \mathrm{d}\vec{x}, \quad i \in \mathbb{N}(n_u), \ j \in \mathbb{N}(N_{\bar{\Omega}}), \tag{33}$$

where $\bar{\mathcal{I}} : \mathbb{N}(n_u) \times \mathbb{N}(N_{\bar{\Omega}}) \to \mathbb{N}(\bar{N})$ 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

$$\bar{\boldsymbol{C}}_{\bar{\mathcal{I}}(i,j),\mathcal{I}(l,k)} = \frac{|\Omega_k|}{|\bar{\Omega}_j|}\delta_{il} I(\Omega_k \subseteq \bar{\Omega}_j), \tag{34}$$

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

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

$$\bar{\boldsymbol{f}}(\boldsymbol{x};\boldsymbol{\mu}) = \bar{\boldsymbol{C}}\boldsymbol{f}(\boldsymbol{x};\boldsymbol{\mu}), \tag{35}$$

such that subdomain conservation can be expressed as

$$\bar{\boldsymbol{C}}\boldsymbol{f}(\boldsymbol{x};\boldsymbol{\mu}) = 0. \tag{36}$$

For the detailed explanation on the derivation of Eqs. (35)–(36), 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 (28) 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

$$\tilde{\tilde{\boldsymbol{x}}}(t;\boldsymbol{\mu}) = \boldsymbol{h}(\boldsymbol{x}^0(\boldsymbol{\mu}) + \boldsymbol{\Phi}\hat{\boldsymbol{x}}(t;\boldsymbol{\mu})), \tag{37}$$

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 ℓ^2-norm, i.e., [20]

$$\hat{\boldsymbol{x}}(\boldsymbol{\mu}) = \underset{\hat{\boldsymbol{z}} \in \mathbb{R}^p}{\arg\min} \left\| \boldsymbol{A}\boldsymbol{f}\left(\boldsymbol{h}(\boldsymbol{x}^0(\boldsymbol{\mu}) + \boldsymbol{\Phi}\hat{\boldsymbol{z}}(\boldsymbol{\mu})); \boldsymbol{\mu}\right)\right\|_2, \tag{38}$$

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

$$\tilde{\tilde{u}}_1 = \max(\epsilon_1, \tilde{u}_1), \tag{39}$$

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 (6) using (7) and (15) to derive the following expression for temperature

$$c_{\mathrm{v}} T = E - \frac{1}{2}(v_j v_j)$$

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

$$\tilde{\tilde{u}}_5 = \max\left(\epsilon_5 + \frac{1}{2\tilde{\tilde{u}}_1}\left[\tilde{u}_2^2 + \tilde{u}_3^2 + \tilde{u}_4^2\right], \tilde{u}_5\right), \tag{40}$$

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 (39) to $\tilde{\boldsymbol{x}}$, and $\tilde{\tilde{\boldsymbol{x}}} = \boldsymbol{h}_5(\tilde{z})$ applies (40) 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].

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 $1/m$

Table 1 Free stream flow conditions for the HiFIRE-1 for run 34 of the CUBRC wind tunnel experiments [22].

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 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 in a minimally intrusive manner. The main design principle behind *Pressio* is that an application only needs to satisfy a minimal application programming interface (API). This consists of exposing, 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 it does 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 and C-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),[22]. 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.

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 Euler time step and scheduled increases in CFL number. The convergence criteria are a reduction in relative residual by 12 orders of magnitude or 25000 pseudo-time steps. Near the baseline parameters, the solver converges in around 15,000 steps, but convergence is slower at lower values of freestream density and velocity, resulting in only 5 orders of magnitude convergence for some cases.

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

[‡]https://github.com/Pressio

Fig. 1 HIFiRE mesh.

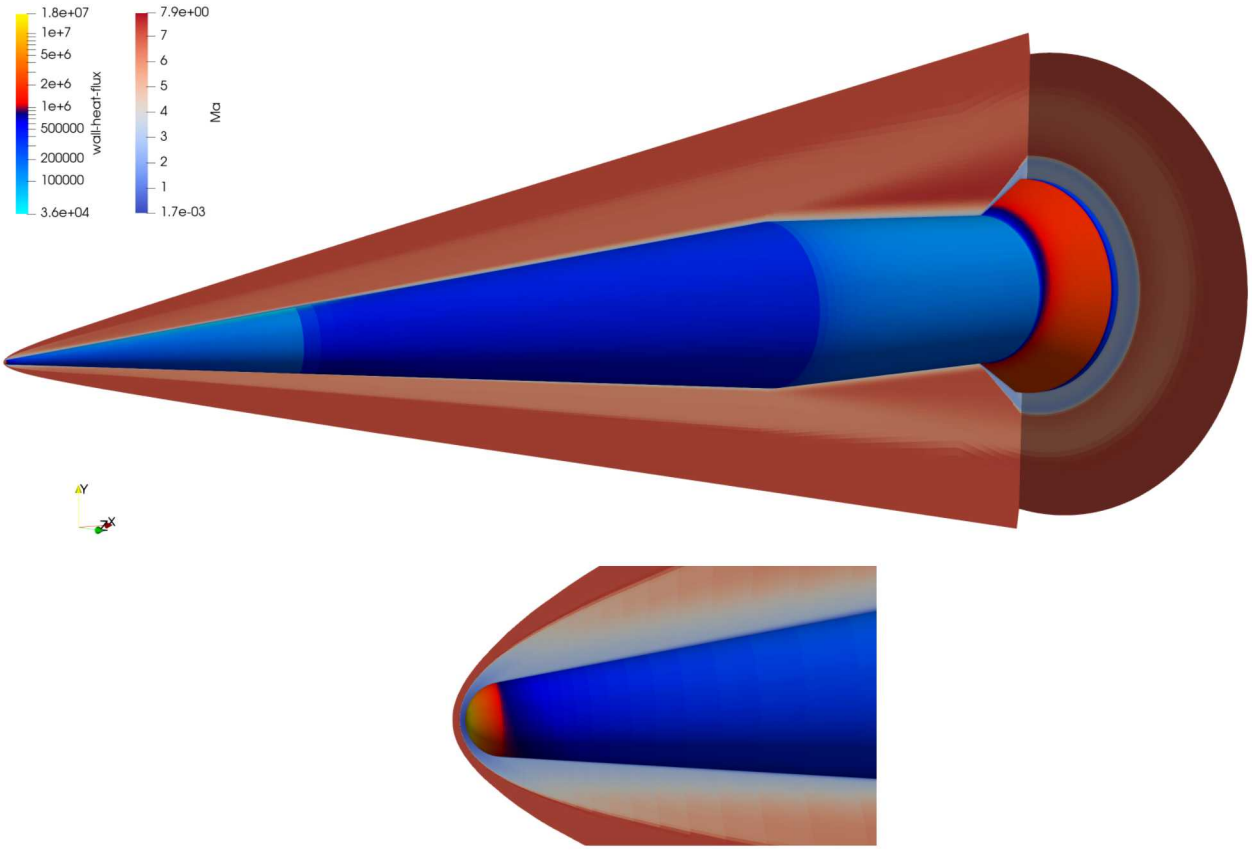


Fig. 2 SPARC simulation of HIFiRE-1 at the freestream condition in table 1. The flow field is colored by Mach number M, and the vehicle surface is colored by wall heat flux q_{wall}.

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 found that the full order model computes heat fluxes similar to that observed by the wind tunnel experiments on most of the vehicle, but underpredicts it at the beginning of the flare. 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.

Reduced-Order Model The snapshot matrix $\boldsymbol{X}_{\text{snap}}$ is formed by FOM solutions with the ith column corresponding 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 defined as

$$\boldsymbol{\Phi} = \boldsymbol{D}_{max}\bar{\boldsymbol{\Phi}}, \tag{41}$$

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 $\boldsymbol{D}_{max}$,

$$\boldsymbol{D}_{max}^{-1}(\boldsymbol{X}_{\text{snap}} - \boldsymbol{x}^0). \tag{42}$$

This normalization is done because SPARC was run with dimensional quantities, as is standard practice for hypersonic CFD codes, especially for cases with non-equilibrium 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 (42) better conditioned than the centered, unscaled snapshot matrix $(\boldsymbol{X}_{\text{snap}} - \boldsymbol{x}^0)$.

Because of the diagonal matrix in (41), 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, the pseudo inverse is easy to compute as

$$\hat{\boldsymbol{x}}(\boldsymbol{\mu}) = (\boldsymbol{\Phi})^+(\boldsymbol{x}(\boldsymbol{\mu}) - \boldsymbol{x}^0) = \bar{\boldsymbol{\Phi}}^T\boldsymbol{D}_{max}^{-1}(\boldsymbol{x}(\boldsymbol{\mu}) - \boldsymbol{x}^0). \tag{43}$$

The selection of the weighting matrix $\boldsymbol{A}$ in (29) 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

$$\boldsymbol{D}_{\mathcal{I}_{(i,j)},\mathcal{I}_{(i,j)}} = \Omega_j, \quad i \in \mathbb{N}(n_u), j \in \mathbb{N}(N_\Omega). \tag{44}$$

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 since 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 penalized boundary condition violations more heavily there. Similar behavior with regard to residual weighting was observed by [23] 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 [23]. 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 [12].

Collocation is implemented by use of a sample mesh [1], 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 conservation 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.

The ROM is solved using a Gauss Newton method instead of the pseudo time stepping used to solve the FOM, as done in [23]. 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 [23, Algorithm 23].

We consider the LSPG, conservative LSPG (C-LSPG), and manifold LSPG (M-LSPG) formulations presented in section III. The non-linear least-squares problems arising from LSPG and M-LSPG are solved via a QR-based Gauss-Newton, while the one stemming from C-LSPG is solved via normal equations. The Gauss-Newton solver used is provided by *Pressio*. In both cases, the Gauss-Newton solvers are run until the relative residual L2 norm falls below 10^{-4} 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 with hyper-reduction, we only consider the approximate conservation

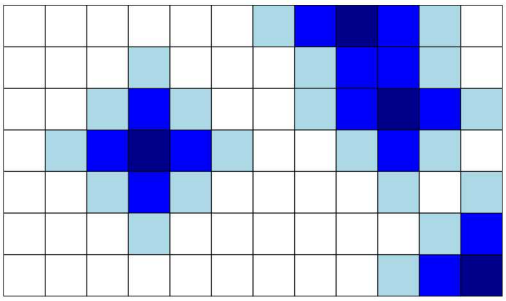


Fig. 3 Schematic showing cells included in a sample mesh. The residual f is sampled at the dark cells. The neighboring cells (blue), and their neighbors (light blue) are included in the sample mesh since computing f in the middle cell requires the states x at all highlighted cells.

constraint (31). 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

$$\mathcal{E}_{\boldsymbol{x}} = \frac{\|\boldsymbol{x}(\boldsymbol{\mu}) - \tilde{\boldsymbol{x}}(\boldsymbol{\mu})\|_2}{\|\boldsymbol{x}(\boldsymbol{\mu})\|_2} \tag{45}$$

where $\boldsymbol{x}(\boldsymbol{\mu})$ and $\tilde{\boldsymbol{x}}(\boldsymbol{\mu})$ are the full state computed with the FOM and some approximation with (28), 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 wall heat flux L2 error, defined as

$$\mathcal{E}_{\boldsymbol{q}_{wall}} = \frac{\|\boldsymbol{q}_{wall}(\boldsymbol{\mu}) - \tilde{\boldsymbol{q}}_{wall}(\boldsymbol{\mu})\|_2}{\|\boldsymbol{q}_{wall}(\boldsymbol{\mu})\|_2} \tag{46}$$

where $\boldsymbol{q}_{wall}(\boldsymbol{\mu})$ and $\tilde{\boldsymbol{q}}_{wall}(\boldsymbol{\mu})$ are vectors containing the heat flux at all wall cell centers computed with the FOM and some approximation with (28), respectively. Wall heat flux was chosen since heating is a key design driver for hypersonic vehicles.

As in [23], 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 $\boldsymbol{\Phi}$,

$$\tilde{\boldsymbol{x}}_{FOM}(\boldsymbol{\mu}) \equiv \boldsymbol{\Phi}(\boldsymbol{\Phi})^{+}(\boldsymbol{x}(\boldsymbol{\mu}) - \boldsymbol{x}^0). \tag{47}$$

This provides an lower bound for the ROM state error, since it is the most accurate representation of the FOM solution $\tilde{\boldsymbol{x}}(\boldsymbol{\mu})$ possible with the basis $\boldsymbol{\Phi}$. However, $\tilde{\boldsymbol{x}}_{FOM}(\boldsymbol{\mu})$ is not necessarily a lower bound for fields derived from the state vector like $\tilde{\boldsymbol{q}}_{wall}(\boldsymbol{\mu})$, so we only compute $\mathcal{E}_{\boldsymbol{x}}$ with $\tilde{\boldsymbol{x}}_{FOM}(\boldsymbol{\mu})$ to compare with ROM solution and ROM initial guess errors.

B. Two parameter ROM

Our first series of numerical experiments demonstrate a ROM parametrized by free stream velocity and density. The parameters for the training data and test data are selected by two independent latin hypercube samplings, 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. Table 2 shows that over 99% of the training set's cumulative statistical energy can be captured with as few as 2 modes. We consider three different ROM dimensions $p = 2, 4, 8$, using the first 2,4, and 8 basis vectors, respectively.

Table 3 shows that the scales of each conserved quantity range over 9 orders of magnitude, a very wide range, highlighting the need to scale snapshots prior to computing POD modes for this case.

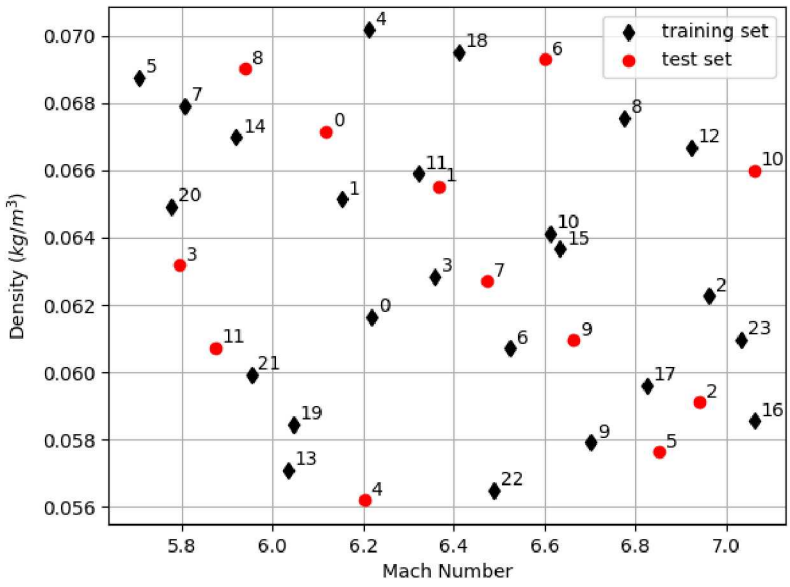


Fig. 4 Training and test data plotted on parameter space and labeled by case number. The freestream density and velocity were both chosen to range from $[0.8\rho_\infty, \rho_\infty]$ and $[0.8v_\infty, v_\infty]$, respectively, where ρ_∞ and v_∞ are the baseline values specified in table 1. The leading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

Mode	Cumulative Statistical Energy
1	0.92210
2	0.99188
3	0.99890
4	0.99971
5	0.99988
6	0.99993
7	0.99995
8	0.99997

Table 2 Cumulative energy for case 1 POD Basis $\tilde{\Phi}$.

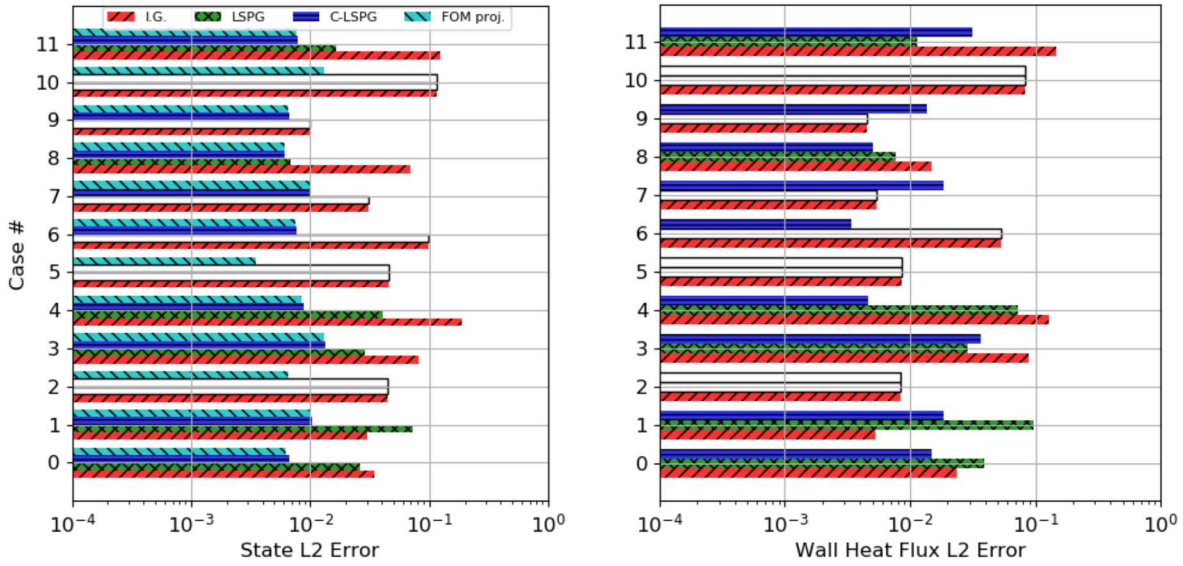
Conserved Variable	Maximum Absolute Value	Units
ρ	0.2695	kg/m^3
ρv_1	97.52	$kg/(m^2 s)$
ρv_2	103.6	$kg/(m^2 s)$
ρv_3	101.7	$kg/(m^2 s)$
ρE	164900	$kgJ/(m^3)$
$\rho\phi_1$	0.006415	$kg^2/(m^2 s)$

Table 3 Case 1 conserved variable scales in D_{max}.

LSPG and conservative LSPG without hyper-reduction A number of different types of ROMs were run on the test set shown in Figure 4 and compared to FOM solutions computed on the same test set. Overall, it appears that C-LSPG is considerably more accurate than LSPG for all three ROM sizes, p, that were tested.

For ROM dimension $p = 2$, Figure 5a shows that the $\mathcal{E}_x$ for C-LSPG is just under 1% for most cases, around 10 times more accurate than the initial guess in most cases. LSPG is similarly accurate for some cases but for case 1 it is actually less accurate than the initial guess. Cases 2 and 10 fail for both LSPG and C-LSPG, due to non-physical initial guesses that have local regions of negative temperature. Case 5 fails due to the solver reaching a state with local regions of negative temperature. Case 7 fails for LSPG but succeeds for C-LSPG.

For ROM dimension $p = 8$, both LSPG and C-LSPG solutions have a lower $\mathcal{E}_x$ than the initial guess and no cases fail. In cases 4, 10, and 11, the $\mathcal{E}_x$ for the C-LSPG solution is around 0.1%, about 100 times smaller than the $\mathcal{E}_x$ of



(a) $\mathcal{E}_{\boldsymbol{x}}$, $p = 2$

(b) $\mathcal{E}_{\boldsymbol{q}_{wall}}$, $p = 2$

(c) $\mathcal{E}_{\boldsymbol{x}}$, $p = 8$

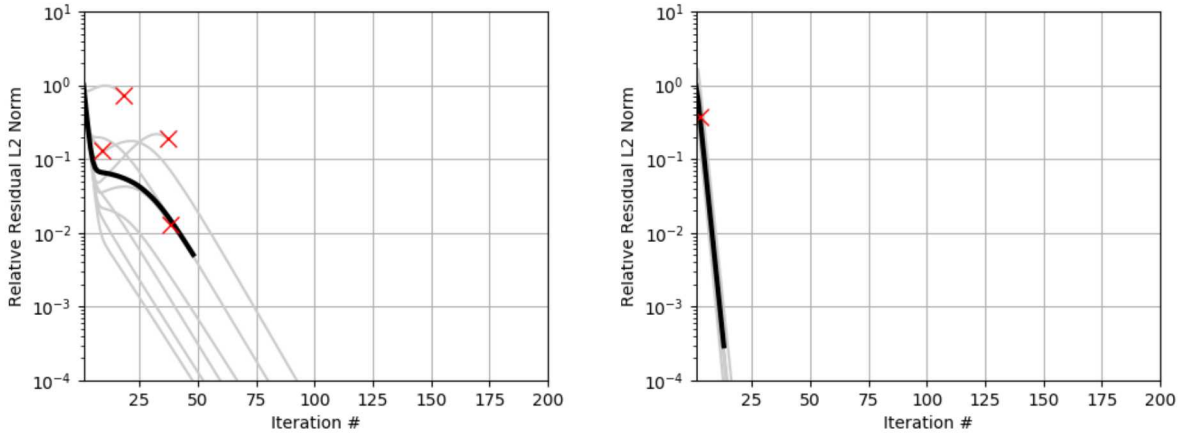
(d) $\mathcal{E}_{\boldsymbol{q}_{wall}}$, $p = 8$

Fig. 5 Relative L2 error of state vector, $\mathcal{E}_{\boldsymbol{x}}$, and wall heat flux vector, $\mathcal{E}_{\boldsymbol{q}_{wall}}$ for different cases and ROM dimensions, p. Case numbers correspond to the labels in Figure 4. FOM proj. refers to the error obtained by projecting the FOM solution $\boldsymbol{x}(\mu)$ on the basis $\boldsymbol{\Phi}$ as in (47). White bars indicate cases that failed. Note that the results for $p = 4$ had errors only slightly larger than those computed for $p = 8$.

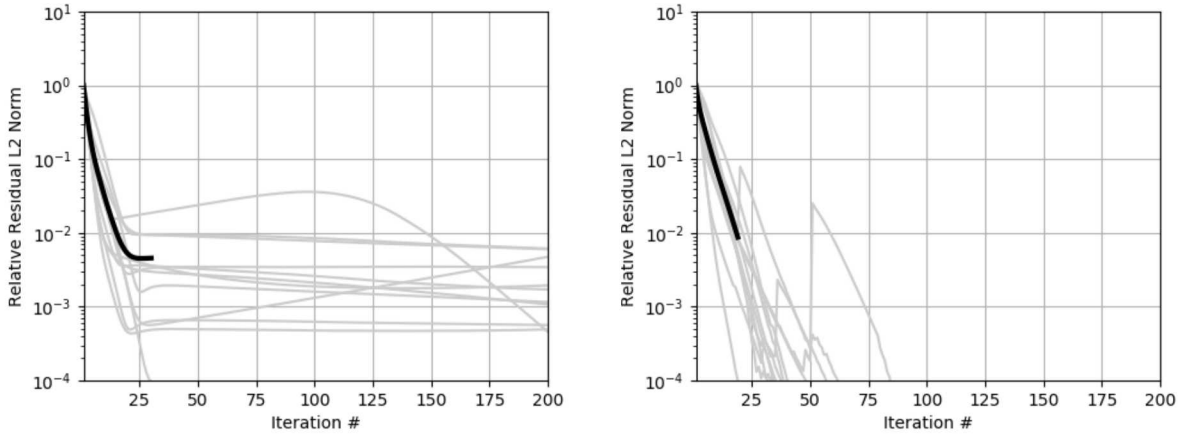
around 10% for the corresponding initial guess. The C-LSPG solution is also around 10 times more accurate than the $p = 2$ solutions. The large improvement in accuracy is due to better approximation offered by the $p = 8$ basis, as shown by the large decrease in FOM projection error when p is increased to 8.

Figures 5a and 5c show that the state error, $\mathcal{E}_{\boldsymbol{x}}$, computed with C-LSPG is very close to the FOM projection error when $p = 2$ and 8. This means that C-LSPG is nearly as accurate as possible for the basis. For $p = 8$, $\mathcal{E}_{\boldsymbol{x}}$ is roughly one order of magnitude lower for C-LSPG than LSPG. Results for $p = 4$ are similar to $p = 8$ and are not displayed in figure 5.

The error for derived quantities such as the wall heat flux error, $\mathcal{E}_{\boldsymbol{q}_{wall}}$, behaves similarly to $\mathcal{E}_{\boldsymbol{x}}$, but can be



(a) LSPG, $p = 2$ **(b) C-LSPG, $p = 2$**

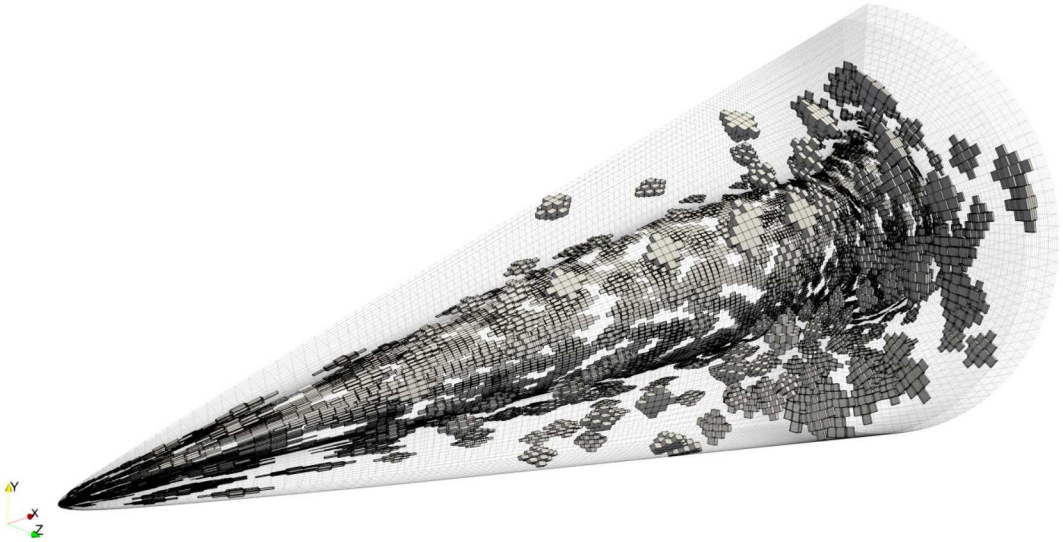


(c) LSPG, $p = 8$ **(d) C-LSPG, $p = 8$**

Fig. 6 Relative residual L2 norms for LSPG and C-LSPG for different ROM dimensions p. Mean relative residual history is plotted in black. Red X's mark iteration on which cases failed.

considerably larger, as shown in Figures 5b and 5d. For both ROM dimensions, $\mathcal{E}_{\boldsymbol{q}_{wall}}$ are up to 1-3% for C-LSPG or as high as 10% for LSPG. In most cases, $\mathcal{E}_{\boldsymbol{q}_{wall}}$ is lower for C-LSPG than LSPG, but there are some exceptions, including cases 1 and 3 for $p = 2$ and case 4 for $p = 8$. Additionally, $\mathcal{E}_{\boldsymbol{q}_{wall}}$ is actually smaller for $p = 2$ than $p = 8$ in some cases. In cases 4,6, and 8, C-LSPG computes a lower value of $\mathcal{E}_{\boldsymbol{q}_{wall}}$ with a lower ROM dimension. This result is counter-intuitive, but is consistent with the fact that although LSPG guarantees a reduction in state error as p is increased, there is no such guarantee for derived quantities like wall heat flux.

In addition to being more accurate, C-LSPG converges in far fewer Gauss-Newton iterations than LSPG, as shown in Figure 6. For ROM dimension $p = 2$, Figure 6a shows the relative residual norm for LSPG reaches 10^{-4} in roughly 50-100 iterations, while Figure 6b shows that C-LSPG converges to the same relative residual norm in only 10-15 iterations. For $p = 8$, the residual norm stops decreasing or increases for some LSPG cases after roughly 20 Gauss-Newton iterations, as shown in Figure 6c. Figure 6d shows that the residual norms for C-LSPG with ROM dimension $p = 8$ typically reached 10^{-4} after around 25-50 Gauss-Newton iterations, with some exceptions. The large jumps in residual like that observed around iteration 50 are due to the solver reaching states that violate one of the conservation constraints in (30). Note that these jumps were not observed for $p = 4$. Additionally, the C-LSPG residuals for $p = 4$ converged at slightly faster rates that those for $p = 8$, but convergence was still slower than $p = 2$. The degradation of nonlinear convergence rates as the ROM dimension is increased is likely due to the increased stiffness of



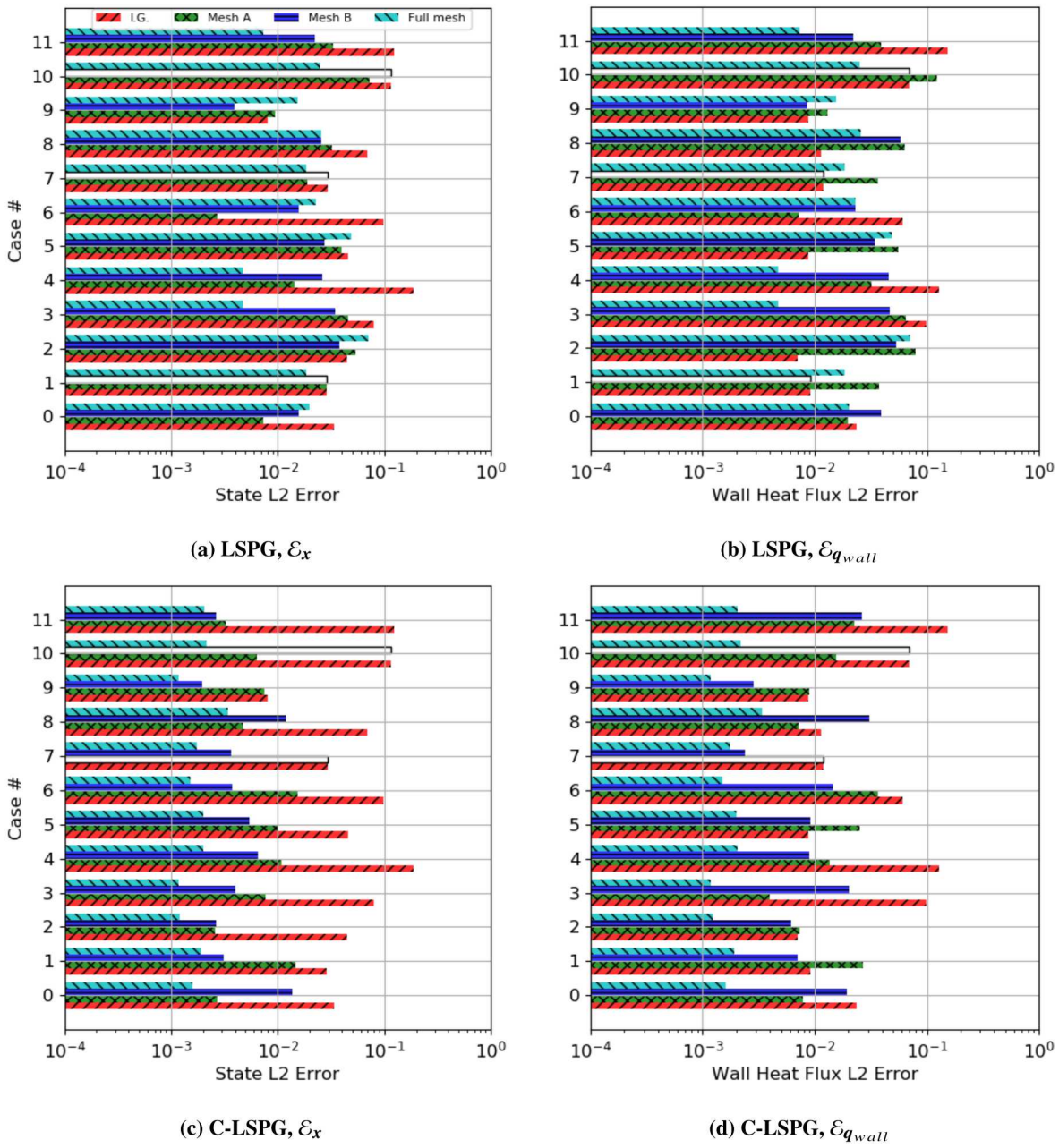
(a) Sample mesh A. The residual is sampled at 2032 randomly selected cells (0.1% of all cells), requiring 49467 cells (2.4% of all cells) to keep neighbors and neighbors of neighbors.

(b) Sample mesh B. The residual is sampled at 813 randomly selected cells (0.04% of all cells), requiring 19901 total cells (0.98% of all cells) to keep neighbors 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.

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.

LSPG and conservative LSPG with hyper-reduction Although the C-LSPG ROM is accurate, it 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. We demonstrate LSPG and C-LSPG with hyper-reduction in the following section, for a single ROM dimension of $p = 4$. This ROM dimension was chosen because it was only slightly less accurate than the $p = 8$ ROMs, but converged in fewer Gauss-Newton iterations. We present results for two sample meshes, which we refer to as sample mesh A and sample mesh B, both of



(a) LSPG, $\mathcal{E}_{\boldsymbol{x}}$ **(b) LSPG, $\mathcal{E}_{\boldsymbol{q}_{wall}}$**

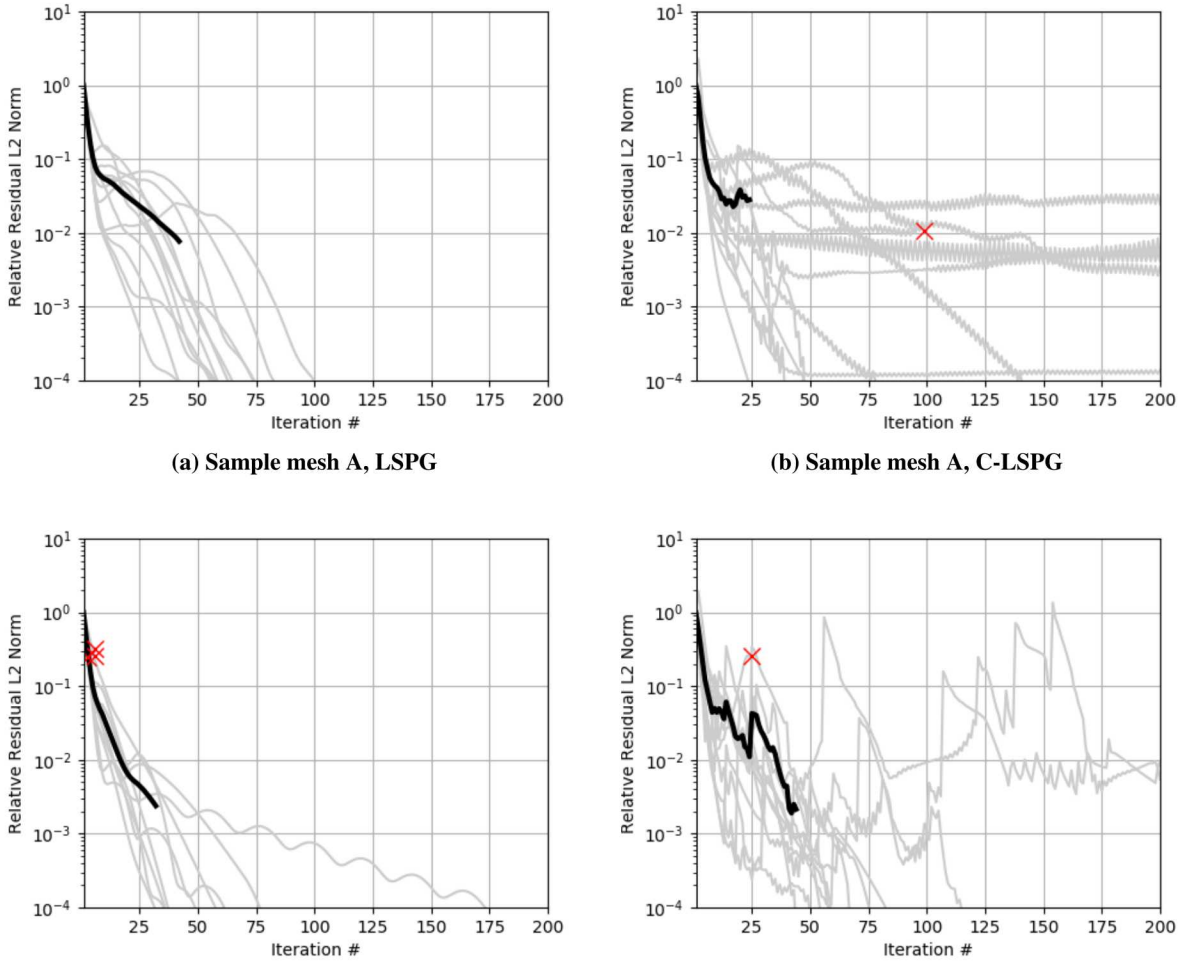
(c) C-LSPG, $\mathcal{E}_{\boldsymbol{x}}$ **(d) C-LSPG, $\mathcal{E}_{\boldsymbol{q}_{wall}}$**

Fig. 8 Relative state and heat flux errors, $\mathcal{E}_{\boldsymbol{x}}$ and $\mathcal{E}_{\boldsymbol{q}_{wall}}$, for hyper-reduced LSPG and conservative LSPG (C-LSPG) computed with ROM dimension $p = 4$ on both sample meshes. Case numbers correspond to the labels in Figure 4. White bars indicate cases that failed.

which are shown in Figure 7. Sample mesh A contains roughly 2.4% of the full mesh, while sample mesh B contains only 1% of the full mesh cells.

As for the full mesh ROM, C-LSPG is more accurate than LSPG, with lower values of $\mathcal{E}_{\boldsymbol{x}}$ and $\mathcal{E}_{\boldsymbol{q}_{wall}}$ in almost all cases, as shown in Figure 8. Recall that hyper-reduced C-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. Despite this, C-LSPG performs very well; although $\mathcal{E}_{\boldsymbol{x}}$ is always larger than that computed for C-LSPG on the full mesh, it is still roughly 1% or less for both sample meshes.

Figures 8b and 8d show that $\mathcal{E}_{\boldsymbol{q}_{wall}}$ is mostly smaller for C-LSPG solutions than LSPG solutions. Additionally, the



(a) Sample mesh A, LSPG

(b) Sample mesh A, C-LSPG

(c) Sample mesh B, LSPG

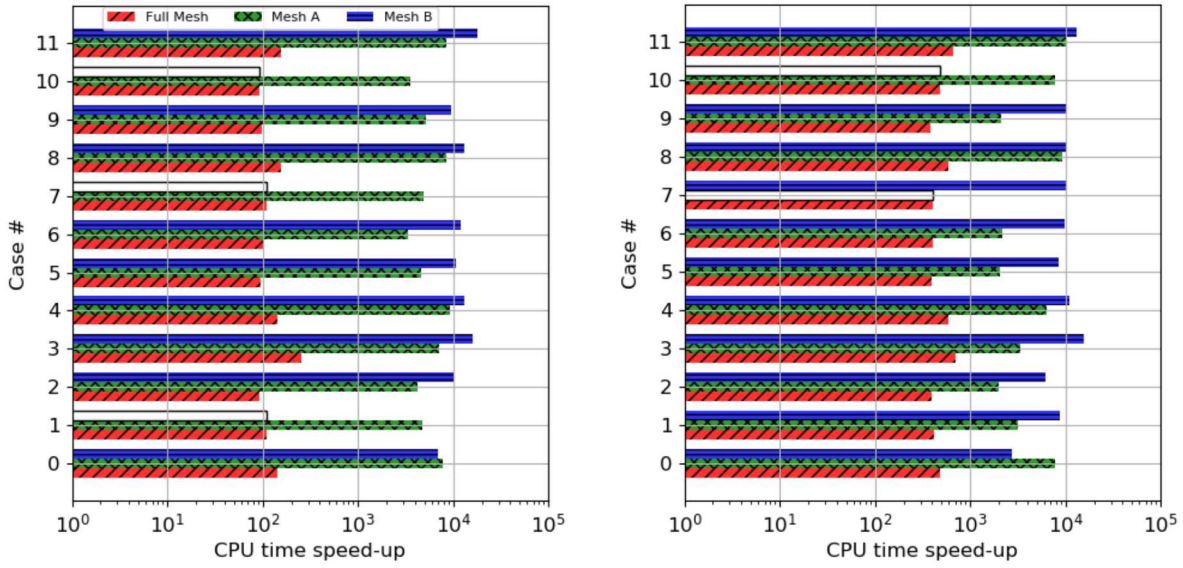
(d) Sample mesh B, C-LSPG

Fig. 9 Relative residual L2 norms for hyper-reduced LSPG and C-LSPG computed with ROM dimension $p = 4$ on both sample meshes. Mean relative residual history is plotted in black. Red X's mark iteration on which cases failed.

hyper-reduced C-LSPG solution is much more accurate than the initial guess for a number of cases, but it is no more accurate than the initial guess in others. Specifically, $\mathcal{E}_{\boldsymbol{q}_{wall}}$ for the hyper-reduced C-LSPG solution is 5 to 10 time smaller than the initial guess error $\mathcal{E}_{\boldsymbol{q}_{wall}}$ for cases 3,4,10 and 11 on one or both sample meshes. Interestingly, all of these cases lie near the edges of the parameter ranges over which the ROM is trained (see figure 4). On the other hand, the initial guess $\mathcal{E}_{\boldsymbol{q}_{wall}}$ is similar or smaller than $\mathcal{E}_{\boldsymbol{q}_{wall}}$ for the C-LSPG solution for cases 1,2, and 5. These cases are among a group of cases for which the initial guess $\mathcal{E}_{\boldsymbol{q}_{wall}}$ is only around 1%, which also includes cases 7, 8, and 9. These cases are mostly well inside the training set used for $\boldsymbol{\Phi}$. This shows that hyper-reduced C-LSPG is most useful near the edge of the parameter space, where surrogate models like the inverse-distance interpolation used to compute the initial guess for the ROM have much larger ($\geq 10\%$) errors.

Interestingly, the ROMs using sample mesh A are not consistently more accurate than those using sample mesh B. This is despite the fact that mesh A more than twice as many cells as mesh B. Mesh A is more robust for LSPG; some LSPG cases with sample mesh B fail. It should be noted that when a mesh coarser than mesh B was tested it was observed to fail on most cases when the solver reaches some $\hat{\boldsymbol{x}}$ corresponding to a non-physical approximate state $\tilde{\boldsymbol{x}}$. This suggests that using a sample mesh with more cells can make LSPG more robust.

Unlike LSPG, only one case fails on each mesh for C-LSPG. Additionally, $\mathcal{E}_{\boldsymbol{x}}$ is lower for C-LSPG solutions on



(a) LSPG **(b) C-LSPG**

Fig. 10 Speed-up of the ROM over full order model for LSPG and C-LSPG. Case numbers correspond to the labels in Figure 4. White bars indicate cases that failed. ROM and FOM were both run on Intel Sandy Bridge processors. The FOM cases were run with 128 cores, while the ROM was 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 [12].

mesh B than mesh A for 7 out of the 12 test cases, despite the fact that mesh B is contained in mesh A. Some of this counter-intuitive behavior might be due to the nonlinear solver convergence issues for C-LSPG with hyper-reduction shown in Figure 9. Unlike the full mesh cases, LSPG on both sample meshes tends to converge faster than C-LSPG, especially for sample mesh A, the larger sample mesh. Figure 9b shows that most residuals converge slowly or stop converging before reaching the target relative residual norm of 10^{-4}. In contrast, only 2 cases fail to reach a relative residual norm of 10^{-4} for C-LSPG on sample mesh B. Overall, it seems that the nonlinear solver is more efficient on smaller sample meshes for LSPG and C-LSPG, when it does not encounter a non-physical state. However, it seems that C-LSPG is more difficult to solve on the sample mesh, at least for the cases considered in this paper.

LSPG and C-LSPG performance To estimate the performance of LSPG and C-LSPG relative to the FOM, we compared the CPU times of the $p = 4$ dimension ROMs with the FOMs for all 12 test cases. Figure 10 shows that the hyper-reduced ROMs are between 2,000 and 10,000 faster than the FOM at the test cases. As in [1] these speed-up estimates include the time needed to compute $\boldsymbol{q}_{wall}$ as well as other wall quantities including 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 size of this speed-up means that thousands 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.

Figure 10 also 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 is also related to 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, which requires residual snapshots taken from full mesh ROMs, may be feasible for steady flow problems.

Before proceeding, it should be emphasized that these speed-ups are just an estimate computed from single runs of the ROM and FOM, each on a fixed number of processors. A more rigorous analysis would be comprised of strong scaling tests for both the FOM and ROM, with speed-ups computed from the *average* CPU times of the peak performance

Case #	Free Stream Velocity (m/s)	Mach Number
0	1518.1	4.97
1	1648.2	5.40
2	1778.3	5.83
3	1908.5	6.25
4	2038.6	6.68
5	2168.7	7.10

Table 4 Training set free stream velocities and corresponding Mach numbers. All other free stream conditions are equal to those in table 1

Case #	Free Stream Velocity (m/s)	Mach Number
0	1583.2	5.19
1	1713.3	5.61
2	1843.4	6.04
3	1973.5	6.46
4	2103.6	6.89

Table 5 Test set free stream velocities and corresponding Mach numbers. All other free stream conditions are equal to those in table 1

FOM and ROM.

Also, recall that the convergence criteria for the FOM are much stricter than those for the ROM. The FOM convergence criteria are relative residual norm of less than 10^{-12} (or 25,000 pseudo time steps) versus a relative residual norm of less than 10^{-4} (or 200 Gauss-Newton iterations) for the ROM. However, even if one only ran the FOM to a relative residual norm of 10^{-4}, it would reduce the number of pseudo time steps to around 7,500; a third to a half of the pseudo time steps required to get to 10^{-12}. Making the rough assumption that wall time is proportional to the number of pseudo timesteps, the ROM is still 600-3,000 times faster than the FOM, which is a sizable speed up.

C. Single parameter ROM with larger parameter range

In section IV.B, a number of ROMs fail when the solver encounters an approximate state $\tilde{\boldsymbol{x}}$ with local non-physical conditions like negative temperatures. The same issue arises when the ROM is trained with fewer snapshots in a given region of parameter space. We would like to maximize the range of parameters over which the ROM is accurate while minimizing the amount of snapshots needed to train the ROM basis. However, it seems that the Gauss-Newton solver is more likely to come across non-physical states when the basis is computed from snapshots that are taken further apart in parameter space. These failures due to non-physical states could be addressed by using a line search or the manifold-LSPG formulation with clipping discussed in section III.D.

The following experiment shows one example of this with a single parameter ROM. The single parameter we consider is the free stream velocity. The training and test parameters are listed in tables 4 and 5, respectively. They are both evenly spaced on an interval of free stream velocities $[0.7v_\infty, v_\infty]$, where v_∞ is the baseline free stream velocity listed in table 1. This is a 50% increase in the velocity range considered in section IV.B, and it results in the leading shock wave rotating approximately 2° over the training parameter range.

The matrices comprising the basis $\boldsymbol{\Phi}$, $\boldsymbol{D}_{max}$ and $\bar{\boldsymbol{\Phi}}$, are computed using the FOM solutions at the training points in table 4. Table 6 shows that over 99% of the training set's cumulative statistical energy can be captured with as few as 2 modes. We set the ROM dimension to $p = 4$ for this study.

Table 7 shows that the scales of each conserved quantity range over 8 orders of magnitude, a very wide range, highlighting the need to scale snapshots prior to computing POD modes for this case.

As in section IV.B, the accuracy of the ROM is measured by the state error $\mathcal{E}_{\boldsymbol{x}}$ and wall heat flux error $\mathcal{E}_{\boldsymbol{q}_{wall}}$ defined in (45) and (46), respectively. The ROM and FOM are computed on the test set in table 5 to compute these errors.

LSPG or C-LSPG both fail on 4 out of the 5 test cases when the solver finds an approximate state $\tilde{\boldsymbol{x}}$ with negative temperatures. ROM solutions can be only be computed for the test case corresponding to the lowest Mach number, $M = 5.19$.

The nonlinear solver can avoid non-physical states by conducting a line search at each Gauss-Newton iteration. Pressio currently supports an (optional) backtracking line search method based on the Armijo rule [24]. The search is stopped if the rule is met, or if the absolute value of the correction is smaller than 10^{-14}. Figure 11a shows that $\mathcal{E}_{\boldsymbol{x}}$ and $\mathcal{E}_{\boldsymbol{q}_{wall}}$ computed by LSPG with a line search are around 4-5 times smaller than the initial guess errors.

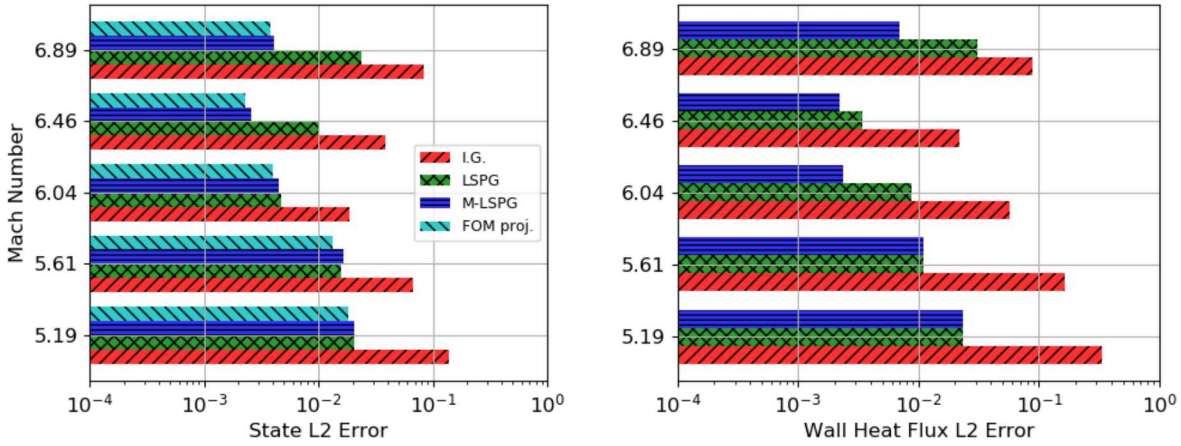
Accuracy can be further improved by using M-LSPG with clipping. The clipping functions (39) and (40) are both

Mode	Cumulative Statistical Energy
1	0.97008
2	0.99284
3	0.99831
4	0.99980
5	1.0
6	1.0

Table 6 Cumulative energy for case 2 POD Basis $\bar{\Phi}$.

Conserved Variable	Maximum Absolute Value	Units
ρ	0.4217	kg/m^3
ρv_1	194.4	$kg/(m^2 s)$
ρv_2	124.3	$kg/(m^2 s)$
ρv_3	108.6	$kg/(m^2 s)$
ρE	261300	$kgJ/(m^3)$
$\rho \phi_1$	0.01929	$kg^2/(m^2 s)$

Table 7 Case 2 conserved variable scales in D_{max}.



(a) $\mathcal{E}_{\boldsymbol{x}}$ **(b)** $\mathcal{E}_{\boldsymbol{q}_{wall}}$

Fig. 11 Relative state and heat flux errors, $\mathcal{E}_{\boldsymbol{x}}$ and $\mathcal{E}_{\boldsymbol{q}_{wall}}$, for LSPG with a line search and M-LSPG, both with ROM dimension $p = 4$. FOM proj. refers to the error obtained by projecting the FOM solution $\boldsymbol{x}(\mu)$ on the basis $\boldsymbol{\Phi}$ as in (47).

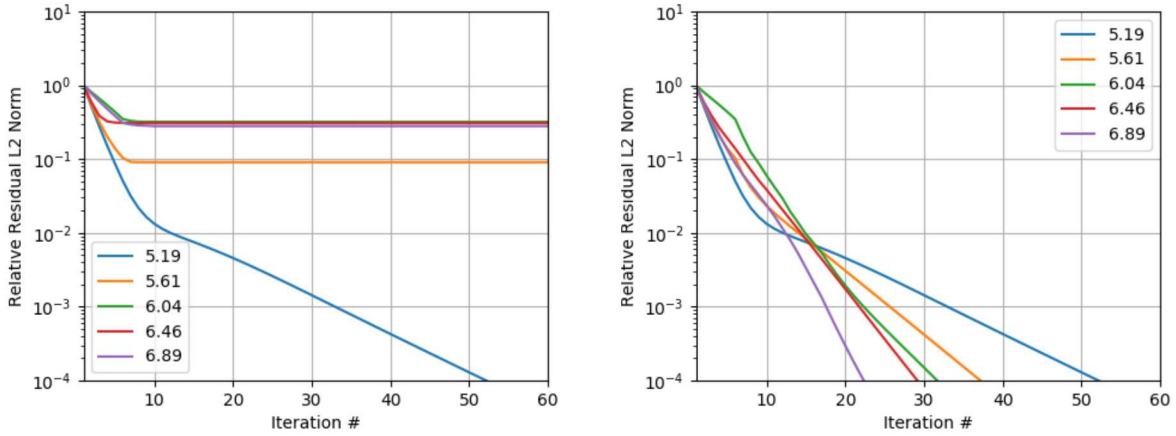
applied with $\epsilon_1 = \epsilon_5 = 10^{-6}$. Note that no regions of negative density were found, only negative temperature had to be clipped implicitly by (40). Figure 11 shows that M-LSPG has similar or smaller state errors than LSPG with the line search for all cases. For the larger Mach number cases, M-LSPG is considerably more accurate than LSPG with the line search. Unlike the line search, $\mathcal{E}_{\boldsymbol{x}}$ for the M-LSPG solution is very close to the FOM projection.

The superior accuracy of M-LSPG is achieved because it can minimize the residual much more than LSPG with line search, as shown in Figure 12. The residual norms are identical for the $M = 5.19$ case, because neither the line search and clipping functions are active for this case. For all other cases, the line search is able to backtrack from solution updates that lead to non-physical states, but the backtracking eventually results in a correction with a magnitude of 10^{-14}, causing the relative residual norm to level out around 0.1 or 0.3 for the $M > 5.19$ test cases. The clipping function (40) keeps M-LSPG away from states with negative temperatures, resulting in residual convergence rates that are even faster than that of the $M = 5.19$ case.

V. Conclusions

In conclusion, LSPG ROMs can compute accurate approximations of conserved quantity fields (the state vector) and derived quantity fields such as wall heat flux for steady-state flows around a hypersonic flight vehicle at a fraction of the cost of the full order model.

The results of section IV.B show that conservative LSPG in particular is accurate and fairly robust when trained with sufficient snapshot data over a parameter range. The 2-parameter C-LSPG ROM state errors are 0.1% and heat



(a) LSPG **(b) M-LSPG**

Fig. 12 Relative residual L2 norms for LSPG with a line search and M-LSPG.

flux errors were at most 1-3%. Hyper-reduction via collocation with random cell sampling maintains a high level of accuracy for the state vector and derived fields like wall heat flux while increasing performance substantially over the full mesh ROM. This is particularly true for C-LSPG, despite the crude approximation of the conservation constraint by collocation. Hyper-reduced ROMs are thousands of times faster than the FOM, which means they could make many-query analyses feasible for large steady state hypersonic flow simulations.

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 off 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 an 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 computing ROMs in some parts of parameter space, and surrogate models in other regions.

Finally, we find 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 line search or 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 that follow up on the results presented in this paper. Firstly, M-LSPG needs to be tested with hyper-reduction and the conservation constraint. Secondly, to apply M-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. Third, although collocation with random cell selection works well, there are likely better hyper-reduction strategies that can improve upon the accuracy and/or performance presented in this paper. The speed-ups obtained for C-LSPG on the full mesh suggest that GNAT hyper-reduction may be feasible for steady state hypersonic flow simulations.

The poor nonlinear convergence rates observed for some LSPG and C-LSPG cases suggests that more work is needed on the LSPG and C-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.

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] 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.

[2] 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.

[3] 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. doi:10.2514/1.C032233, URL https://doi.org/10.2514/1.C032233.

[4] 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. doi:10.2514/1.A32592, URL https://doi.org/10.2514/1.A32592.

[5] 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. doi:10.2514/1.A32947, URL https://doi.org/10.2514/1.A32947.

[6] 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. doi:10.2514/1.A32716, URL https://doi.org/10.2514/1.A32716.

[7] 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. doi:10.2514/6.2010-7930.

[8] 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. doi:10.2514/1.J050802.

[9] 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. doi:10.2514/1.J051094.

[10] 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.

[11] 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.

[12] 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.

[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, Vol. 1. doi: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. doi:10.2514/6.2017-4407, URL https://arc.aiaa.org/doi/abs/10.2514/6.2017-4407.

[22] 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. doi:10.2514/1.38338, URL https://doi.org/10.2514/1.38338.

[23] 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.

[24] Nocedal, J., and Wright, S. J., *Numerical Optimization*, 2nd ed., Springer, New York, NY, USA, 2006.