

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, Wyoming 82801

Micah Howard‡ and Jeffrey A. Fike‡

Sandia National Laboratories, Albuquerque, New Mexico 87185-0828

and

Kevin T. Carlberg§

University of Washington, Seattle, Washington 98195

<https://doi.org/10.2514/1.J059785>

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 (FOM). 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 FOM 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 Hypersonic International Flight Research Experimentation 1 (HIFIRE-1) in a three-dimensional, turbulent Mach 7.1 flow. This paper shows the ability of the ROM to significantly reduce computational costs while maintaining high levels of accuracy in computed quantities of interest.

Nomenclature

A	=	least-squares Petrov–Galerkin weight matrix
b	=	vector of nonlinear boundary conditions
\bar{b}	=	vector of prescribed boundary conditions
C	=	speed of sound
\bar{C}	=	conservation constraint matrix
c_p	=	specific heat at constant pressure
c_v	=	specific heat at constant volume
D	=	cell volume scaling matrix
D_{\max}	=	basis scaling matrix
d	=	spatial dimension
$ds(\mathbf{x})$	=	integration with respect to a boundary
E	=	total energy per unit mass
e	=	fluid internal energy
e_i	=	cell face i
F_i	=	inviscid flux vector
$F_{i,t}^{\text{turb}}$	=	turbulence model inviscid flux of scalar t in Cartesian direction i
f	=	residual vector
\tilde{f}	=	approximate residual vector

G_i	=	viscous flux vector
$G_{i,t}^{\text{turb}}$	=	turbulence model viscous flux of scalar t in Cartesian direction i
g_i	=	flux vector for conserved quantity i
h	=	internal enthalpy
\mathcal{h}	=	clipping function
I	=	indicator function
\mathcal{I}	=	mapping from conservation-law index and control-volume index to degree of freedom
I	=	identity matrix
\mathcal{M}	=	finite-volume mesh
$\bar{\mathcal{M}}$	=	decomposed finite-volume mesh
Ma	=	Mach Number
N	=	number of states
\mathbb{N}	=	set of integers
N_e	=	number of faces in finite-volume mesh
N_Ω	=	number of control volumes
$N_{\bar{\Omega}}$	=	number of spatial subdomains
\mathbf{n}	=	normal vector
n_t	=	number of turbulent transport equations
n_u	=	number of conserved quantities
n_μ	=	number of parameters
P	=	fluid pressure
P_r	=	sampling matrix
Pr	=	Prandtl number
p	=	number of columns in reduced-basis matrix Φ
q_i	=	heat flux in Cartesian direction i
R	=	specific gas constant
$\mathbb{R}^{n \times m}$	=	space of $n \times m$ real-valued matrices
\mathcal{S}	=	nonlinear trial manifold
S	=	constant in Sutherland formula
\mathcal{S}	=	source vector
S_t	=	source term for turbulent scalar t
s_i	=	source term for conserved quantity i
T	=	gas temperature
t	=	time

Presented as Paper 2020-0104 at the AIAA SciTech 2020 Forum, Orlando, FL, January 6–10, 2020; received 12 May 2020; revision received 31 August 2020; accepted for publication 4 October 2020; published online 5 February 2021. Copyright © 2021 by the American Institute of Aeronautics and Astronautics, Inc. Under the copyright claimed herein, the U.S. Government has a royalty-free license to exercise all rights for Governmental purposes. All other rights are reserved by the copyright owner. All requests for copying and permission to reprint should be submitted to CCC at www.copyright.com; employ the eISSN 1533-385X to initiate your request. See also AIAA Rights and Permissions www.aiaa.org/randp.

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

†Principal Technical Lead.

‡Aerosciences, PO Box 5800, MS 0825. Member AIAA.

§Affiliate Associate Professor, Applied Mathematics and Mechanical Engineering, Lewis Hall #201, Box 353925.

\mathbf{U}	=	conserved quantity vector
u_i	=	conserved quantity i
\tilde{u}_i	=	i th conserved quantity field for approximate state vector $\tilde{\mathbf{x}}$
\tilde{u}_i	=	i th conserved quantity field for approximate state vector $\tilde{\mathbf{x}}$
v_j	=	fluid velocity in j th Cartesian direction
\mathbf{X}_{snap}	=	snapshots matrix
\mathbf{x}	=	state vector
$\tilde{\mathbf{x}}$	=	approximate state vector from nonlinear manifold
$\hat{\mathbf{x}}$	=	generalized coordinate vector
$\tilde{\mathbf{x}}$	=	approximate state vector
\mathbf{x}	=	position in spatial domain Ω
$\hat{\mathbf{z}}$	=	dummy generalized coordinate vector
Γ	=	boundary of domain Ω
$\bar{\Gamma}_i$	=	boundary of subdomain i
Γ_i	=	boundary of control volume i
γ	=	boundary of subdomain ω
γ	=	specific heat ratio
δ_{ij}	=	Dirac delta function for $i = j$
\mathcal{E}	=	set of cell faces in finite-volume mesh
\mathcal{E}_i	=	set of cell faces in control volume i
$\bar{\mathcal{E}}_i$	=	set of faces in subdomain i
$\mathcal{E}_{Q_{\text{wall}}}$	=	integrated heat flux error
$\mathcal{E}_{\mathbf{x}}$	=	state error
ϵ_i	=	clipper tolerance for conserved quantity i
κ	=	gas thermal conductivity
λ	=	bulk viscosity
μ	=	viscosity
$\boldsymbol{\mu}$	=	vector of parameters
ρ	=	fluid density
τ_{ij}	=	viscous stress tensor
Φ	=	reduced-basis matrix
$\bar{\Phi}$	=	unscaled reduced-basis matrix
Φ_r	=	reduced-basis matrix for residual snapshots
ϕ_t	=	turbulent transport scalar t
Ψ	=	test basis matrix
Ω	=	spatial domain
$\bar{\Omega}_i$	=	subdomain i
Ω_i	=	control volume i
ω	=	general subdomain of domain Ω

Subscripts

FOM	=	full-order model projection
wall	=	quantity on a wall
+	=	strictly positive

Superscripts

FV	=	finite-volume approximation
g	=	finite-volume source contribution
s	=	finite-volume flux contribution
0	=	initial condition

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 nonequilibrium 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 (2D) 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, although 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 because 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.

Despite these advantages, there are some challenges to applying projection-based ROMs to hypersonic flows. The main difficulty is that projection-based ROMs can struggle to accurately approximate discontinuities such as shockwaves when they move in time or with respect to parameter perturbations. This is particularly true for popular basis choices like POD modes [10]. A number of projection-based ROMs with modified minimization statements and/or basis computation strategies have been presented to improve the accuracy of ROMs for shock-dominated fluid flows. Lucia et al. apply a hybrid ROM–FOM approach where the computational domain is decomposed into subdomains with and without shockwaves [11]. ROMs are solved in the subdomains without shockwaves, whereas the FOM is applied in regions with shockwaves. The technique was demonstrated on a 2D supersonic flow around a bluff body. Abgrall and Crisovan [10] and Abgrall et al. [12] use a reduced basis comprising snapshots and utilize L^1 projection to make their ROMs robust for shock-dominated flows. Their approach has been demonstrated on one-dimensional (1D) unsteady and 2D steady transonic flows. Nair and Balajewicz also leverage a basis comprising snapshots, but transform the snapshots such that the shockwaves are in the same position [13]. This approach demonstrated improved accuracy over LSPG and the L^1 minimization approach mentioned earlier for several 1D and 2D steady supersonic flows.

All of these novel approaches for ROMs of shock-dominated flows are somewhat difficult to implement, and have not been demonstrated on large-scale three-dimensional (3D) flows. The hybrid ROM–FOM approach requires highly intrusive modifications to the CFD solver to enable domain decomposition. The L^1 minimization approach requires extensive modifications to the nonlinear solver/optimizer that solves the ROM governing equations. Nair's snapshot transformation approach and others like it require a mapping function, which might be difficult to construct for arbitrarily complex computational domains. In addition, snapshot transformation requires deforming meshes and interpolating between meshes, both of which can be difficult to implement and are not readily available for many large-scale CFD solvers. We hope to exploit some aspects of these approaches in follow-on work to this study.

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

In light of the difficulties with other methods described above, this work focuses on the least-squares Petrov–Galerkin (LSPG) projection [14], leaving extensions to more novel ROM approaches to future work. The accuracy and stability of LSPG have been demonstrated on large-scale problems in CFD [15–17], and its flexible optimization-based formulation that readily admits integration of constraints that enforce conservation laws over subdomains [18]. Specifically, we present the application of LSPG to hypersonic CFD simulations. In doing so, we make the following contributions:

- 1) We present, to the best of our knowledge, the first application of projection-based ROMs to large-scale hypersonic CFD simulations.
- 2) We provide a detailed comparison of the accuracy and predictive performance of two ROM methods.
- 3) We demonstrate the impact of satisfying conservation laws and physical constraints on the convergence of the reduced systems.
- 4) We highlight the advantages and pitfalls of LSPG with an outlook to several new potential research directions.

The paper begins with an overview of the FOM in Sec. II, followed by an overview of projection-based ROM techniques in Sec. III, and a series of numerical experiments with a CFD simulation of a Hypersonic International Flight Research Experimentation 1 (HIFIRE-1) wind tunnel test in Sec. IV. Finally, Sec. V offers some conclusions and directions for future work.

II. Full-Order Model: Finite-Volume Discretization of Hypersonic Aerodynamic Flows

A. Governing Equations and Discretization

In this paper, we consider high-Mach external aerodynamics with enthalpy not sufficiently high to drive dissociation of the gas we consider (air). Hence, we solve the perfect gas, compressible Reynolds-averaged Navier–Stokes (RANS) equations for conserved quantities:

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

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_i is a set of scalars that belong to turbulent transport equations and n_t is the number of turbulent transport equations. For the full set of governing equations, please consult Appendices A and B and/or [19].

We consider steady-state solutions of the RANS equations, which can be expressed as the solution of the following system:

$$\mathbf{f}(\mathbf{x}; \boldsymbol{\mu}) = \mathbf{0} \quad (2)$$

where \mathbf{f} is the residual vector, $\mathbf{x} \in \mathbb{R}^N$ is the state vector, and $\boldsymbol{\mu}$ is the vector of system parameters. The governing equations are discretized with a cell-centered FV method [20,21]. Therefore, the state vector \mathbf{x} is composed of each conserved quantity at each cell center, whereas the residual vector \mathbf{f} is composed of the discretized conservation laws (the governing equations) at each cell center. A more detailed derivation of Eq. (2) is included in Appendix C.

B. 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 nonequilibrium effects, it should be noted that nonequilibrium 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_μ , 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 FOM is too computationally expensive to solve enough times to obtain reasonable uncertainty estimates. Such cases demand a method for *approximately* solving the FOM while retaining high levels of accuracy. We now present a method that 1) computes a low-dimensional representation of the state using a linear subspace, 2) computes a model for the resulting latent state that exactly satisfies the physical conservation laws over the entire mesh, and 3) removes nonphysical flow features from the state representation using nonlinear clipper functions.

III. Reduced-Order Modeling

A. Least-Squares Petrov–Galerkin Projection

Classical projection-based ROMs compute an approximate solution $\tilde{\mathbf{x}} \approx \mathbf{x}$ from an affine function:

$$\tilde{\mathbf{x}}(t; \boldsymbol{\mu}) = \mathbf{x}^0(\boldsymbol{\mu}) + \boldsymbol{\Phi} \hat{\mathbf{x}}(t; \boldsymbol{\mu}) \quad (3)$$

where $\boldsymbol{\Phi} \in \mathbb{R}^{N \times p}$ is the reduced-basis matrix of dimension $p \leq N$ and $\hat{\mathbf{x}} \in \mathbb{R}^p$ denotes the generalized coordinates. This basis can be computed in a variety of ways during the offline stage, e.g., eigenmode analysis, POD [22], or the reduced-basis method [23,24]. Typically, $\boldsymbol{\Phi}$ is orthonormal, and $\boldsymbol{\Phi}^T \boldsymbol{\Phi} = \mathbf{I}$. 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 $\mathbf{x} \leftarrow \tilde{\mathbf{x}}$ into the FOM steady-state equations (2), and subsequently minimizes residual in a weighted ℓ^2 norm, i.e.,

$$\hat{\mathbf{x}} = \arg \min_{\hat{\mathbf{x}} \in \mathbb{R}^p} \|\mathbf{A} \mathbf{f}(\mathbf{x}^0(\boldsymbol{\mu}) + \boldsymbol{\Phi} \hat{\mathbf{x}}; \boldsymbol{\mu})\|_2 \quad (4)$$

where $\mathbf{A} = \mathbf{I}$, for example.

The first-order optimality conditions for Eq. (4) correspond to a projection equation of the form

$$\boldsymbol{\Psi}^T \mathbf{A} \mathbf{f}(\mathbf{x}^0(\boldsymbol{\mu}) + \boldsymbol{\Phi} \hat{\mathbf{x}}; \boldsymbol{\mu}) = \mathbf{0} \quad (5)$$

for which the optimal choice of test basis $\boldsymbol{\Psi} \in \mathbb{R}^{N \times p}$ is

$$\boldsymbol{\Psi} = \mathbf{A} \frac{\partial \mathbf{f}}{\partial \mathbf{x}} \Big|_{\tilde{\mathbf{x}}} \boldsymbol{\Phi} \quad (6)$$

This is in contrast to a Galerkin ROM, for which $\boldsymbol{\Psi} = \boldsymbol{\Phi}$. The Petrov–Galerkin projection is better suited for systems with asymmetric Jacobian matrices $\partial \mathbf{f} / \partial \mathbf{x}$, most notably advection-dominated systems, including Navier–Stokes solvers for practical aerodynamic applications.

The nonlinearity of \mathbf{f} requires evaluating \mathbf{f} and the corresponding Jacobian $\partial \mathbf{f} / \partial \mathbf{x}$ for each nonlinear solver iteration. The computational cost of this scales with N , which limits the computational efficiency of the ROM. To ensure that the ROM incurs an N -independent operation count, the weighting matrix \mathbf{A} should be sparse in the sense that it has a small number of nonzero columns. In this case, one can set $\mathbf{A} = (\mathbf{P}_r \boldsymbol{\Phi}_r)^+ \mathbf{P}_r$ and $\mathbf{A} = \mathbf{P}_r$ in the case of gappy POD and collocation, respectively. Here, $\mathbf{P}_r \in \{0, 1\}^{n_{p,r} \times N}$ denotes a sampling matrix comprising selected rows of the $N \times N$ identity matrix, whereas $\boldsymbol{\Phi}_r \in \mathbb{R}^{N \times p_r}$ is a p_r -dimensional reduced-basis matrix constructed for the residual \mathbf{f} . Employing the gappy POD approximation results in the Gauss-Newton with approximated tensors (GNAT) ROM [15].

B. Conservative LSPG Projection

As proposed in Ref. [18], we now modify the LSPG ROM for steady simulations [Eq. (4)] by enforcing conservation on the decomposed mesh as nonlinear equality constraints. In particular, conservative LSPG (C-LSPG) projection for steady simulations computes a solution $\hat{\mathbf{x}}$ that satisfies

$$\begin{aligned} & \underset{\hat{\mathbf{z}} \in \mathbb{R}^p}{\text{minimize}} && \|A\mathbf{f}(\mathbf{x}^0(\boldsymbol{\mu}) + \Phi\hat{\mathbf{z}}; \boldsymbol{\mu})\|_2 \\ & \text{subject to} && \bar{\mathbf{C}}\mathbf{f}(\mathbf{x}^0(\boldsymbol{\mu}) + \Phi\hat{\mathbf{z}}; \boldsymbol{\mu}) = \mathbf{0} \end{aligned} \quad (7)$$

where $\bar{\mathbf{C}} \in \mathbb{R}_+^{N_{\hat{\mathbf{z}}} n_u \times N}$ is a matrix that maps the residual \mathbf{f} to the conservation violations for all n_u conserved quantities on $N_{\hat{\mathbf{z}}}$ spatial subdomains, although we only consider conservation violations over the entire mesh in this paper. The matrix $\bar{\mathbf{C}}$ is derived and defined in Appendix D.

The first-order optimality conditions for Eq. (4) now correspond to two coupled projection equations [18]:

$$\Psi^T [A\mathbf{f}(\mathbf{x}^0(\boldsymbol{\mu}) + \Phi\hat{\mathbf{x}}; \boldsymbol{\mu}) + \bar{\mathbf{C}}^T \boldsymbol{\lambda}] = \mathbf{0} \quad (8)$$

$$\bar{\mathbf{C}}\mathbf{f}(\mathbf{x}^0(\boldsymbol{\mu}) + \Phi\hat{\mathbf{x}}; \boldsymbol{\mu}) = \mathbf{0} \quad (9)$$

where $\boldsymbol{\lambda} \in \mathbb{R}^p$ denote Lagrange multipliers and the test basis Ψ is the same as in Eq. (6).

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

$$\bar{\mathbf{C}}\tilde{\mathbf{f}}(\mathbf{x}^0(\boldsymbol{\mu}) + \Phi\hat{\mathbf{x}}; \boldsymbol{\mu}) = \mathbf{0} \quad (10)$$

where $\tilde{\mathbf{f}}$ is the approximate residual vector, which is $\tilde{\mathbf{f}} = \mathbf{P}_r^T \mathbf{P}_r \mathbf{f}$ and $\tilde{\mathbf{f}} = \Phi_r(\mathbf{P}_r \Phi_r)^+ \mathbf{P}_r \mathbf{f}$ in the case of collocation and gappy POD, respectively. Reference [18] contains additional details on conservative LSPG projection, including sufficient conditions for feasibility of the associated optimization problems, and a posteriori error bounds.

C. Manifold Least-Squares Petrov–Galerkin Projection

One of the shortcomings of the affine linear trial subspace used to compute the approximate state $\tilde{\mathbf{x}}$ in Eq. (3) is that there exists some $\hat{\mathbf{x}}$ such that $\tilde{\mathbf{x}}$ will contain some nonphysical local phenomena such as regions of negative density or temperature. Reference ([25] Sec. VI.E) shows that ensuring that $\tilde{\mathbf{x}}$ does not have nonphysical local flow features can significantly improve the robustness of LSPG and Galerkin ROMs.

In this paper, we use a nonlinear *trial manifold* that approximates FOM states without nonphysical local features. We consider $\tilde{\mathbf{x}} \approx \mathbf{x}$ of the form

$$\tilde{\mathbf{x}}(t; \boldsymbol{\mu}) = \mathbf{h}(\mathbf{x}^0(\boldsymbol{\mu}) + \Phi\hat{\mathbf{z}}(t; \boldsymbol{\mu})) \quad (11)$$

where $\hat{\mathbf{z}} \in \mathcal{S}$ and $\mathcal{S} := \{\mathbf{h}(\mathbf{x}^0(\boldsymbol{\mu}) + \Phi\hat{\mathbf{z}}) | \hat{\mathbf{z}} \in \mathbb{R}^p\}$ denotes the nonlinear trial manifold from the extrinsic view. Here $\mathbf{h}(\tilde{\mathbf{x}}) \in \mathbb{R}^N$ denotes the clipping function, which comprises a nonlinear mapping from a potentially nonphysical linear affine subspace $\hat{\mathbf{x}} \in \mathbb{R}^p$ to a manifold on which quantities like density and temperature only take physical, nonnegative values. This can be viewed as a relatively simple modification of LSPG; more complicated nonlinear functions can also be considered as in [26].

Similarly to Sec. III.A, steady manifold LSPG (M-LSPG) substitutes the approximation $\mathbf{x} \leftarrow \tilde{\mathbf{x}}$ into the FOM steady-state Eqs. (2), and subsequently minimizes residual in a weighted ℓ^2 norm, i.e., [26]

$$\hat{\mathbf{x}}(\boldsymbol{\mu}) = \arg \min_{\hat{\mathbf{z}} \in \mathbb{R}^p} \|A\mathbf{f}(\mathbf{h}(\mathbf{x}^0(\boldsymbol{\mu}) + \Phi\hat{\mathbf{z}}); \boldsymbol{\mu})\|_2 \quad (12)$$

The first-order optimality conditions for Eq. (12) correspond to a projection equation of the form

$$\Psi^T A\mathbf{f}(\mathbf{h}(\mathbf{x}^0(\boldsymbol{\mu}) + \Phi\hat{\mathbf{x}}; \boldsymbol{\mu})) = \mathbf{0}, \quad \Psi = A \left. \frac{\partial \mathbf{f}}{\partial \mathbf{x}} \right|_{\mathbf{h}(\tilde{\mathbf{x}})} \left. \frac{\partial \mathbf{h}}{\partial \tilde{\mathbf{x}}} \right|_{\tilde{\mathbf{x}}} \Phi \quad (13)$$

Note that the test basis for M-LSPG also includes the Jacobian of the clipping function. This implies that one should choose \mathbf{h} to be differentiable with respect to $\tilde{\mathbf{x}}$; however, we consider a piecewise differentiable \mathbf{h} without issue in this study.

As for LSPG in Eq. (7), constraints can be applied to the M-LSPG minimization statement (12), resulting in conservative manifold LSPG (CM-LSPG). Hyper-reduced constraints can also be applied to CM-LSPG as for C-LSPG in Eq. (10).

The clipping function $\mathbf{h}(\tilde{\mathbf{x}})$ considered in this paper is designed specifically for the case of a single-specie calorically perfect gas; clipping functions for flows with thermal and/or chemical nonequilibrium will be investigated in subsequent works to enable ROMs for higher speed/enthalpy flight regimes.

The perfect gas clipping function $\mathbf{h}(\tilde{\mathbf{x}})$ we consider is designed to enforce fluid density $\rho > 0$ and temperature $T > 0$ in the flowfield represented by $\tilde{\mathbf{x}}$. The density field \tilde{u}_1 is computed by

$$\tilde{u}_1 = \max(\epsilon_1, \tilde{u}_1) \quad (14)$$

where $\epsilon_1 > 0$ is some number that should be very small relative to the freestream density and \tilde{u}_1 is the density field from $\tilde{\mathbf{x}}$. The expression $T > 0$ is derived from the definition of the total energy per unit mass E for a perfect gas as follows:

$$c_v T = E - \frac{1}{2} (v_j v_j) \quad (15)$$

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

$$\tilde{u}_5 = \max\left(\epsilon_5 + \frac{1}{2\tilde{u}_1} [\tilde{u}_2^2 + \tilde{u}_3^2 + \tilde{u}_4^2], \tilde{u}_5\right) \quad (16)$$

Note the presence of \tilde{u}_1 , since the density clipping function must be applied first. Therefore, $\mathbf{h}(\tilde{\mathbf{x}})$ is of the form $\mathbf{h}_5(\mathbf{h}_1(\tilde{\mathbf{x}}))$, where $\tilde{\mathbf{z}} = \mathbf{h}_1(\tilde{\mathbf{x}})$ applies Eq. (14) to $\tilde{\mathbf{x}}$, and $\tilde{\mathbf{x}} = \mathbf{h}_5(\tilde{\mathbf{z}})$ applies Eq. (16) to $\tilde{\mathbf{z}}$. 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 [26]. Finally, it should be noted that a variant of the clipper function with a smooth first derivative was also considered and tested but was not found to have any impact on ROM stability or accuracy for the cases considered.

IV. Numerical Experiments

The results presented below have been obtained using two codes being developed at Sandia National Laboratories, namely, Sandia Parallel Aerodynamics and Reentry Code (SPARC) and Pressio.** SPARC is a compressible CFD code focused on aerodynamics and aerothermodynamics problems. It solves the compressible Navier–Stokes and RANS equations on structured and unstructured grids using a cell-centered FV discretization scheme [19]. Its target use cases are transonic flows to support gravity bomb analyses and hypersonic flows for reentry vehicle analyses. SPARC also solves the transient heat equation and associated equations for nondecomposing 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.

**<https://github.com/Pressio>.

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 Application Programming Interface (API). The main design principle behind Pressio is that an application only needs to expose, for a given state \mathbf{x} , time t , and parameters $\boldsymbol{\mu}$, the velocity vector $\mathbf{f}(\mathbf{x}, t; \boldsymbol{\mu})$ and the action of the Jacobian matrix $\partial \mathbf{f}(\mathbf{x}, t; \boldsymbol{\mu}) / \partial \mathbf{x}$. 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, because it involved the creation of a new adapter class and no changes to the original SPARC code. We remark that although 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, M-LSPG, and CM-LSPG on a SPARC simulation of a wind tunnel test of the HiFiRE-1 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) [27]. The corresponding freestream conditions are listed in Table 1. Additionally, turbulent transition is modeled by tripping the boundary layer 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 nonzero, the vehicle is modeled with the mesh shown in Fig. 1, which discretizes half of the flowfield 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 $n_u = 6$ because 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 Courant–Friedrichs–Lewy (CFL) number. The convergence criteria are a reduction in relative residual by five orders of magnitude or 25,000 pseudo-time steps. Near the baseline parameters, the solver converges in around 9000 steps, but convergence is slower at lower values of freestream density and velocity.

Figure 2 shows the flowfield at several flow conditions, including 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 shockwave 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 FOM 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 [28]. This is a difficult region to predict because flow separation at the

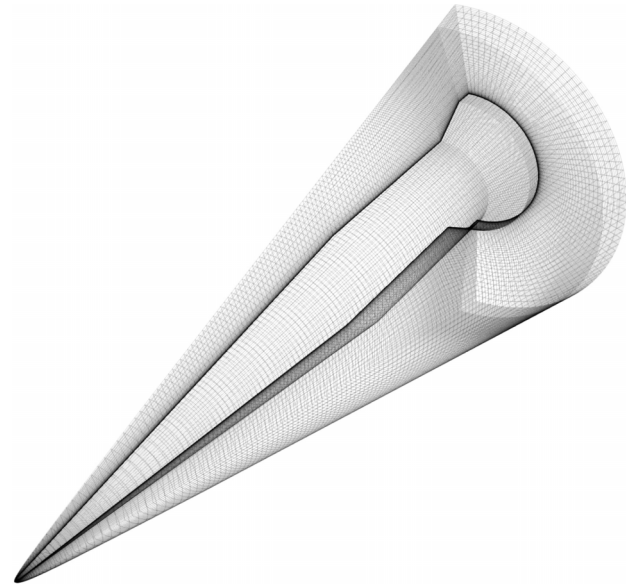


Fig. 1 HiFiRE-1 mesh.

cylinder-flare intersection is highly sensitive and is not well modeled by RANS turbulence models.

C. Reduced-Order Model

In this work, we focus on POD-based model reduction. Current work is ongoing to explore alternative mappings, which is left for future publications. To compute the POD modes for the ROM, we first assemble the snapshot matrix \mathbf{X}_{snap} where the i th column corresponds to an FOM solution \mathbf{x}^i at parameter values $\boldsymbol{\mu}_i$. The reference state $\mathbf{x}^0(\boldsymbol{\mu})$ is simply the mean of all snapshots. Note that this is one relatively straightforward and convenient choice for the reference state; other choices may prove better, and will be the subject of future studies. It should be pointed out that the snapshots mean does not have to be physically or statistically meaningful. Therefore, our choice of $\mathbf{x}^0(\boldsymbol{\mu})$ is nothing more than a convenient reference point in phase space to center our snapshot data.

The basis $\boldsymbol{\Phi}$ is then defined as

$$\boldsymbol{\Phi} = \mathbf{D}_{\max} \bar{\boldsymbol{\Phi}} \quad (17)$$

where $\mathbf{D}_{\max} \in \mathbb{R}^{N \times N}$ is a diagonal matrix with the maximum absolute value of each conserved quantity along the main diagonal. These maxima are computed over all training snapshots. The matrix $\bar{\boldsymbol{\Phi}} \in \mathbb{R}^{N \times p}$ is composed of POD modes computed from the centered snapshot matrix normalized by \mathbf{D}_{\max} . The POD modes are the columns of the left singular vector matrix \mathbf{U} corresponding to the p leading singular values,

$$\mathbf{U} \boldsymbol{\Sigma} \mathbf{V}^T = \mathbf{D}_{\max}^{-1} (\mathbf{X}_{\text{snap}} - \mathbf{x}^0) \quad (18)$$

The normalization by \mathbf{D}_{\max} is done because SPARC is run with dimensional quantities, as is standard practice for hypersonic CFD codes, especially for cases with nonequilibrium thermochemical effects. Normalizing snapshot data before computing POD modes increases numerical robustness. This is because reducing the range of variable scales makes the normalized, centered snapshot matrix (18) better conditioned than the centered, unscaled snapshot matrix $(\mathbf{X}_{\text{snap}} - \mathbf{x}^0)$.

Because of the diagonal matrix in Eq. (17), the basis $\boldsymbol{\Phi}$ is not orthonormal ($\boldsymbol{\Phi}^T \boldsymbol{\Phi} \neq \mathbf{I}$), so projections need to be done with the Moore–Penrose pseudo-inverse of $\boldsymbol{\Phi}$. Since \mathbf{D}_{\max} is diagonal we have

$$\hat{\mathbf{x}}(\boldsymbol{\mu}) = (\boldsymbol{\Phi})^+ (\mathbf{x}(\boldsymbol{\mu}) - \mathbf{x}^0) = \bar{\boldsymbol{\Phi}}^T \mathbf{D}_{\max}^{-1} (\mathbf{x}(\boldsymbol{\mu}) - \mathbf{x}^0) \quad (19)$$

Table 1 Freestream flow conditions for the HiFiRE-1 for run 34 of the CUBRC wind tunnel experiments [27]

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 m ⁻¹

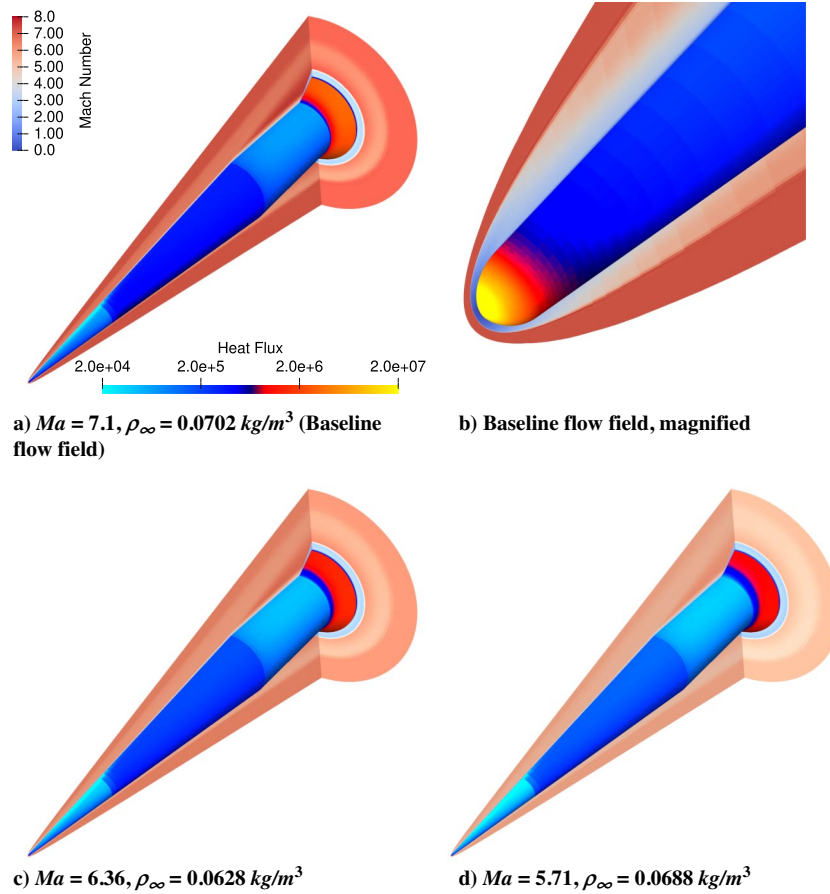


Fig. 2 SPARC simulation of HIFiRE-1. The flowfield is colored by Mach number Ma , and the vehicle surface is colored by wall heat flux q_{wall} . The other freestream properties are those listed in Table 1.

The selection of the weighting matrix A in Eq. (4) is crucial for the accuracy and speed of LSPG. We set $A = D \in \mathbb{R}^{N \times N}$, where $D \in \mathbb{R}^{N \times N}$ is defined as

$$D_{\mathcal{I}_{(i,j)}, \mathcal{I}_{(i,j)}} = |\Omega_j|, \quad i \in \mathbb{N}(n_u), j \in \mathbb{N}(N_{\Omega}) \quad (20)$$

This is a diagonal matrix whose elements correspond to the size of each control volume $|\Omega_k|$. It is found that this choice of A vastly improves the convergence rate of Gauss–Newton iteration, along with accuracy of the ROM solution. This choice of A makes the LSPG residual equivalent to that defined for the FOM. One possible reason for the improved convergence observed when applying 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. Because 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 freestream 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, whereas increased weight on the inflow cells will penalize boundary condition violations more heavily there. Similar behavior with regard to residual weighting was observed by [16] for parametric LSPG applied to steady compressible flows.

Hyper-reduction is done by collocation using $A = P_r D \in \mathbb{R}^{n_r \times N}$, as in [16]. 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 of the computational domain (inflow, vehicle wall, symmetry plane, and outflow) as suggested in [14].

Collocation is implemented by use of a sample mesh [15], in which the steady residual 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 FV scheme, this means that the sample mesh includes collocation cells (dark blue), their neighboring cells (blue), and the neighbors of those neighbors (light blue), as shown in Fig. 3 for a two-dimensional structured mesh.

We consider the LSPG, manifold LSPG (M-LSPG), conservative LSPG (C-LSPG), and conservative manifold LSPG (CM-LSPG) formulations presented in Sec. III. The initial guess for the ROM is computed by an inverse distance interpolation of the projected training data $\hat{x}^i = (\Phi)^+(x^i - x^0)$, specifically ([16], Algorithm 23). The choice of a relatively accurate initial guess combined with projection onto a reduced basis allows us to solve the steady-state equations directly, rather than using pseudo-time stepping for a nonlinear solver as is done for the FOM, as demonstrated in [16]. Foregoing the use of

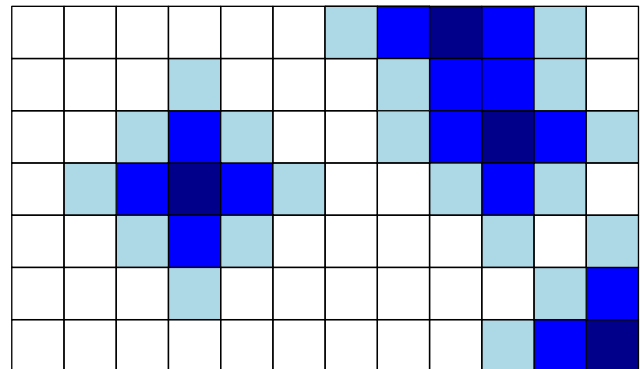


Fig. 3 Schematic showing cells included in a sample mesh.

pseudo-time stepping as a nonlinear solver leads to a considerable reduction in the number of residual evaluations relative to the ROM and vastly simplifies our ROM implementation. The nonlinear least-squares problems arising from LSPG is solved via a QR-based Gauss–Newton, whereas 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. Note that 200 iterations are a relatively large number of iterations selected to prevent the ROM nonlinear solver from running infinitely long.

In this study we only consider global conservation constraints, meaning that the conservation constraint is applied to the entire mesh, leading to six constraints, one for each conserved quantity. Multiple constraints on subdomains, as proposed by Ref. [18], are not considered in this study. For M-LSPG and CM-LSPG, the clipping functions (14) and (16) are applied with $\epsilon_1 = \epsilon_5 = 10^{-6}$. Several tests run with $\epsilon_1 = \epsilon_5 = 10^{-8}$ showed little impact on performance and accuracy. Note that no regions of negative density were found; only negative temperatures had to be clipped implicitly by Eq. (16). For C-LSPG and CM-LSPG with hyper-reduction, we only consider the approximate conservation constraint (10). Reference [18] 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 L^2 error, defined as

$$\mathcal{E}_x = \frac{\|\mathbf{x}(\boldsymbol{\mu}) - \tilde{\mathbf{x}}(\boldsymbol{\mu})\|_2}{\|\mathbf{x}(\boldsymbol{\mu})\|_2} \quad (21)$$

where $\mathbf{x}(\boldsymbol{\mu})$ and $\tilde{\mathbf{x}}(\boldsymbol{\mu})$ are the full state computed with the FOM and some approximation with Eq. (3), respectively. Although this metric does not have a physical interpretation, it provides a means of verifying the LSPG ROM, because the state L^2 error should decrease as $\tilde{\mathbf{x}}$ is increased. Note that the state L^2 error is very close to the L^2 error of conserved energy, ρE , since ρE is much larger in magnitude than the other conserved quantities. The vector $\tilde{\mathbf{x}}(\boldsymbol{\mu})$ is usually the ROM solution, but we also compute $\tilde{\mathbf{x}}(\boldsymbol{\mu})$ for other states as well.

Secondly, we compute the integrated wall heat flux error, defined as

$$\mathcal{E}_{Q_{\text{wall}}} = \frac{|Q_{\text{wall}}(\boldsymbol{\mu}) - \tilde{Q}_{\text{wall}}(\boldsymbol{\mu})|}{|Q_{\text{wall}}(\boldsymbol{\mu})|} \quad (22)$$

where $Q_{\text{wall}}(\boldsymbol{\mu})$ and $\tilde{Q}_{\text{wall}}(\boldsymbol{\mu})$ are the integrals of heat flux at the wall computed with the FOM and some corresponding approximation with Eq. (3), respectively. Integrated wall heat flux was chosen to validate the ROM accuracy because heating is a key design driver for hypersonic vehicles. Other quantities of interest (QoIs) including surface pressure coefficient and skin friction drag were also computed and found to have similar or smaller errors than heat flux. We omit extensive comparisons of these errors for brevity; however, the surface pressure coefficient error is presented and discussed for one case in Sec. IV.

As in [16], 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 Φ , which is a form of surrogate model. This means that the comparison between the ROM solution and initial guess serves two purposes. Firstly, it shows the error reduction achieved by the ROM solver. Secondly, it provides a comparison between the ROM and a relatively inexpensive surrogate model. Therefore, this comparison shows where projection-based ROMs are necessary in parameter space. More extensive studies considering different initial guesses and comparing the ROM with a number of different surrogate models are ongoing and will be presented in future work.

Finally, the state error is also computed for the projection of the FOM solution on the basis Φ as

$$\tilde{\mathbf{x}}_{\text{FOM}}(\boldsymbol{\mu}) \equiv \Phi(\Phi)^+ (\mathbf{x}(\boldsymbol{\mu}) - \mathbf{x}^0) + \mathbf{x}^0 \quad (23)$$

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

Our numerical experiments demonstrate different ROMs parameterized by freestream velocity and density. The freestream density and velocity were 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. As in [16], we select 24 training points in parameter space. The 24 training and 12 test data points in the parameter space are selected with a Latin hypercube sampling, and shown in Fig. 4.

The matrices \mathbf{D}_{max} and $\tilde{\Phi}$ are computed using the FOM solutions at the training points in Fig. 4. Figure 5 shows the cumulative statistical energy for different basis sizes. Over 99% of the training set's cumulative statistical energy can be captured with as few as two modes. We consider three different ROM dimensions $p = 1, 2, 4$, using the first 1, 2, and 4 basis vectors, respectively. Table 2 shows the maximum absolute value of each conserved quantity over the training snapshots. The scales of each conserved quantity vary over nine orders of magnitude, a very wide range, highlighting the need to scale snapshots before computing POD modes for this case.

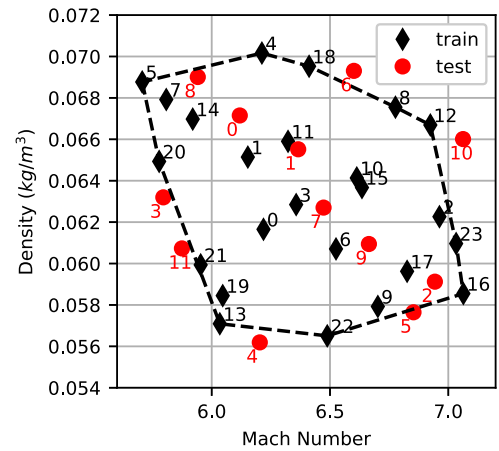


Fig. 4 Training and test data plotted on parameter space and labeled by case number.

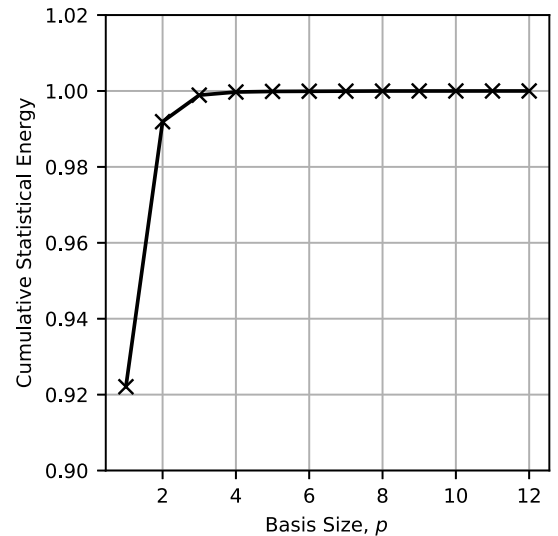


Fig. 5 Cumulative energy for the POD basis considered.

Table 2 Case 1 conserved variable scales in D_{\max}

Conserved variable	Maximum absolute value	Unit
ρ	0.2695	kg/m^3
ρv_1	97.52	$\text{kg}/(\text{m}^2 \cdot \text{s})$
ρv_2	103.6	$\text{kg}/(\text{m}^2 \cdot \text{s})$
ρv_3	101.7	$\text{kg}/(\text{m}^2 \cdot \text{s})$
ρE	164,900	$\text{kJ}/(\text{m}^3)$
$\rho \phi_1$	0.006325	$\text{kg}^2/(\text{m}^2 \cdot \text{s})$

The generalized coordinates $(\Phi)^+(\mathbf{x}(\mu) - \mathbf{x}^0)$ are observed to vary nonlinearly with respect to the parameters ρ_∞ and v_∞ . Specifically, the coordinate for the first mode varies linearly, the second is slightly nonlinear, the third is quadratic with respect to v_∞ , and the fourth is cubic with respect to v_∞ . Plots are provided in the supplemental materials.

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, M-LSPG, C-LSPG, and CM-LSPG ROMs, and the corresponding FOM solutions. Figure 6 shows state errors \mathcal{E}_x [see Eq. (21)] obtained for M-LSPG and CM-LSPG. For the sake of brevity, in the figure we do not report LSPG and C-LSPG because they give equivalent results to M-LSPG and CM-LSPG, respectively, when the clipping functions are inactive. Table 3 shows the test cases for which LSPG or C-LSPG failed and the clipper was active.

The different ROM types exhibit differences in accuracy that become clearer as the ROM dimension p is increased. For $p = 1$, the initial guess has errors ranging from 3 to about 18%, whereas M-LSPG and CM-LSPG offer an improvement, with errors varying between 1 and 4%. The largest differences in error between the ROMs and the initial guess occur for test cases just outside the edge of the training set, including cases 4, 10, and 11 (see Fig. 4). This shows that the ROM is far better at extrapolation than the interpolation scheme using for the initial guess. It should be mentioned that the ROM accuracy is expected to degrade for parameters further from the training set because the basis will struggle to approximate the state far from the training set. Finally, Table 3 shows that LSPG and C-LSPG with $p = 1$ both fail for case 10 due to a nonphysical initial guess with local regions of negative temperature. This issue does not arise for M- and CM-LSPG because the clipper removes the nonphysical features.

Increasing the ROM dimension to $p = 2$ leads to increases in accuracy, but not necessarily the robustness of LSPG and C-LSPG. The error \mathcal{E}_x for CM-LSPG is around 1% or less for most cases, which is roughly 3–10 times more accurate than the initial guess. Note that CM-LSPG is more accurate than M-LSPG for every test case

Table 3 Case numbers for which the ROMs fail for different bases (case numbers correspond to the labels in Fig. 4)

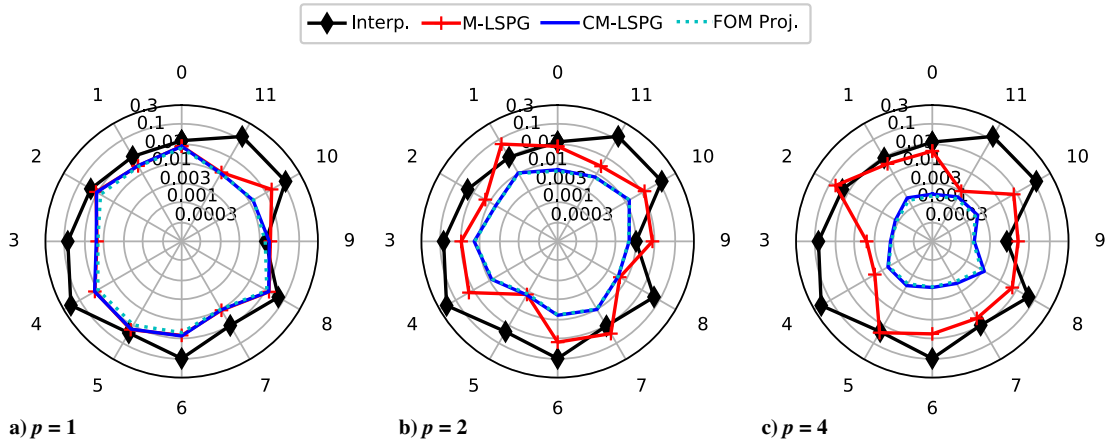
ROM type	$p = 1$	$p = 2$	$p = 4$
LSPG	10	2, 5, 6, 7, 9, 10	None
C-LSPG	10	2, 5, 10	None

except 5. However, we note an increase in the number of cases for which LSPG and C-LSPG fail relative to $p = 1$. Table 3 shows that LSPG fails for half of the tests cases considered. Cases 2 and 10 fail due to nonphysical 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 three failures, cases 2, 5, and 10. One possible explanation for the increase in failures when going from $p = 1$ to $p = 2$ is that increasing the ROM dimension resolves finer scale structure in the solution. Although this typically helps accuracy, for problems with discontinuities such as shockwaves, adding higher-order modes to the basis can lead to spurious oscillations near the shockwave. When these oscillations are large enough, they can lead to nonphysical features. For the test cases considered in this study, negative temperatures show up in a thin region just upstream of the shock near the flange. Interestingly, this region no longer contains negative temperatures for $p = 4$.

For ROM dimension $p = 4$, no cases fail for LSPG or C-LSPG solutions. Therefore, the clipping functions are not active, so M-LSPG and CM-LSPG are equivalent to LSPG and C-LSPG, respectively. LSPG/M-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/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 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}_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, because 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 .

Figure 7 shows a summary of the results obtained for wall heat flux error $\mathcal{E}_{Q_{\text{wall}}}$ [see Eq. (22)] and reveals a different behavior than the one observed for the \mathcal{E}_x . For a given case, $\mathcal{E}_{Q_{\text{wall}}}$ can be considerably larger than \mathcal{E}_x , and there are more cases for which the ROMs do not

**Fig. 6** Relative L^2 error \mathcal{E}_x computed for different ROM dimensions p . Case numbers correspond to the labels in Fig. 4.

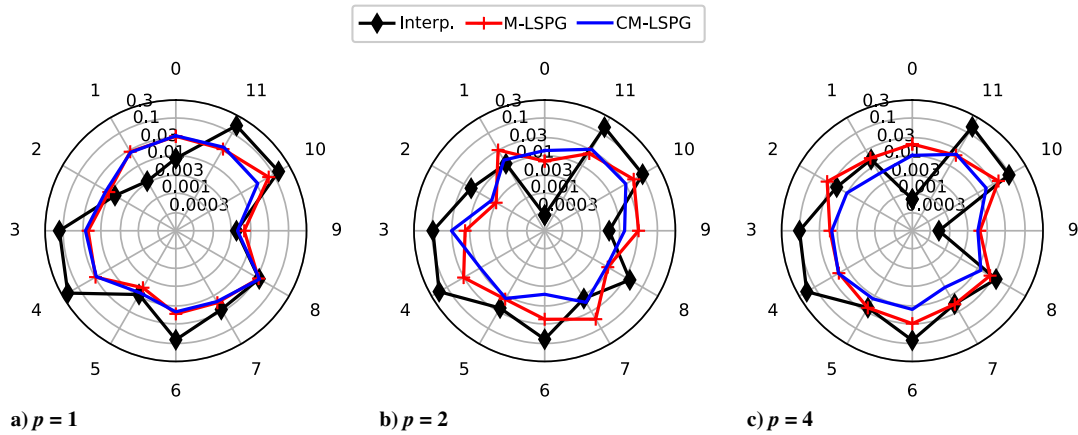


Fig. 7 Relative heat flux error $\mathcal{E}_{Q_{\text{wall}}}$ computed for different ROM dimensions p . Case numbers correspond to the labels in Fig. 4.

offer much or any improvement in accuracy over the initial guess. These larger errors arise for a few reasons. Firstly, \mathcal{E}_x is global quantity, and therefore monotonic convergence of \mathcal{E}_x does not guarantee a monotonic reduction in the local state error for regions such as the vehicle boundary layer. Secondly, heat flux is computed from the temperature field, which from Eq. (15) is itself a nonlinear function of the conserved variables. This means that the temperature field will include errors from the energy, momentum, and density fields. Finally, heat flux is proportional to the temperature gradient. Numerically, this is computed with a finite difference over cell data, in which the difference in temperatures of neighboring cells is divided by the small distance between cell centers, potentially amplifying errors in the temperature field. For further discussion and demonstration of the convergence of a scalar quantity of interest versus state errors with basis size, please consult [29] (Sec. 5).

The improvement in accuracy of the integrated heat flux seems to exhibit a dependence on the input parameters μ . Figure 8 compares $\mathcal{E}_{Q_{\text{wall}}}$ 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 just outside the edges of the training data set. Figure 7 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 $\mathcal{E}_{Q_{\text{wall}}}$ relatively accurately for all the test cases considered. For all ROM dimensions, $\mathcal{E}_{Q_{\text{wall}}}$ is roughly 1–4% for CM-LSPG and M-LSPG. In most cases, $\mathcal{E}_{Q_{\text{wall}}}$ is lower for CM-LSPG than M-LSPG. Additionally, $\mathcal{E}_{Q_{\text{wall}}}$ is actually smaller for $p = 2$ than $p = 4$ in some cases. In several cases, CM-LSPG computes a lower value of $\mathcal{E}_{Q_{\text{wall}}}$ with a lower ROM dimension. This result is counterintuitive, but is consistent with the fact that although LSPG guarantees a monotonic reduction in the global error metric \mathcal{E}_x as p is increased, there is no such guarantee for error of derived field or quantities of interest. This is especially true for integrated wall heat flux due to its dependence on the wall-normal gradient of the temperature field. A reduction in L^2 state error with p does not guarantee a reduction in the state gradient, or the gradient of a derived field. Additionally, the temperature gradient from which the heat flux is computed is highly localized in space, so the decrease in the global error \mathcal{E}_x is a result of error reductions in other regions of the spatial domain.

Along with integrated heat flux, the relative error for the local heat flux on the vehicle surface was computed. Overall, local heat flux errors are observed to be of a similar magnitude to the wall heat flux integral. Surface plots of the heat flux errors are omitted for the full mesh because of their similarity to the results obtained for the hyper-reduced ROMs, which are presented in the next subsection.

In addition to being more accurate and robust, CM-LSPG converges in far fewer Gauss–Newton iterations than LSPG, as shown in Fig. 9. For ROM dimension $p = 2$, Fig. 9a shows that the relative residual norm for LSPG reaches 10^{-5} in roughly 60–110 iterations,

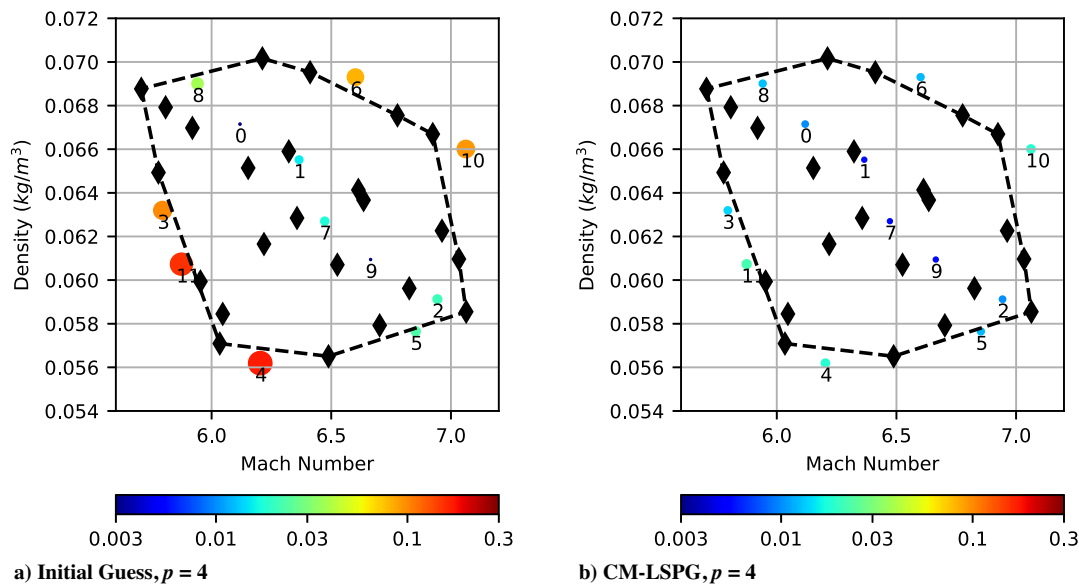


Fig. 8 Parameter space plot with symbols sized and colored by $\mathcal{E}_{Q_{\text{wall}}}$.

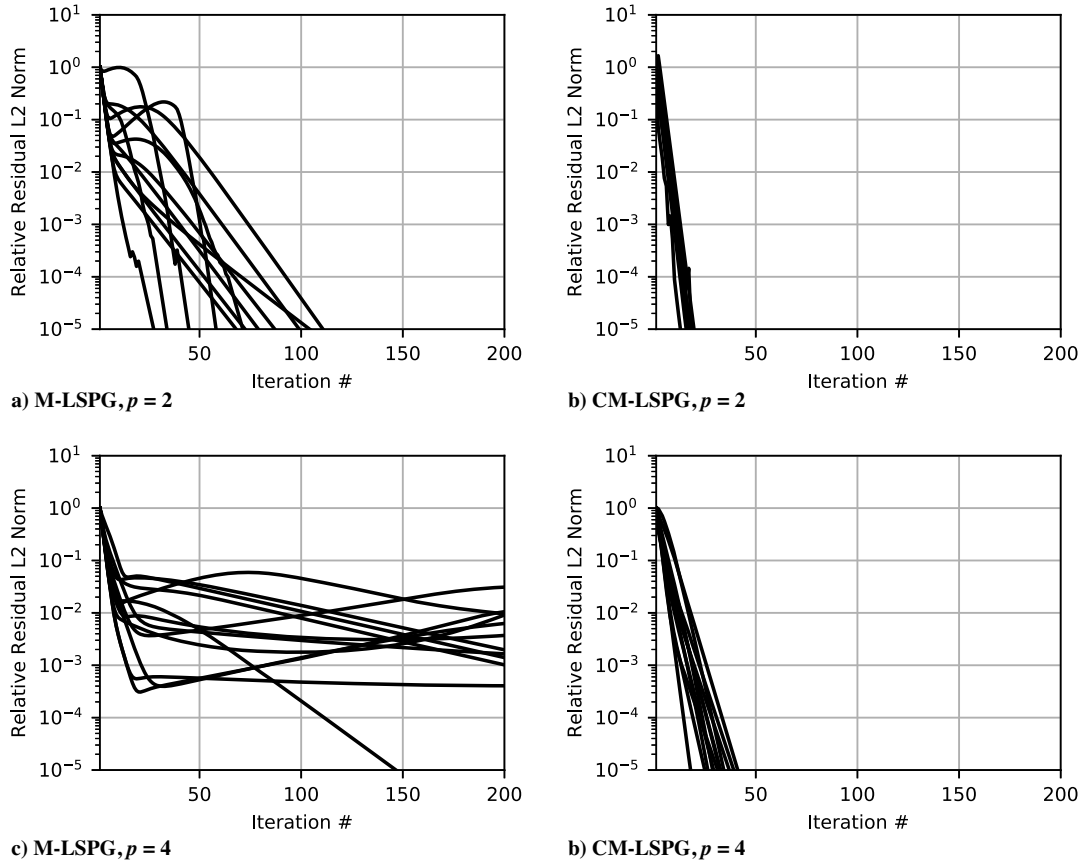


Fig. 9 Relative residual L^2 norms for LSPG and CM-LSPG for different ROM dimensions p .

whereas Fig. 9b 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 Fig. 9c. 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 in part due to a slight increase in the condition number of the LSPG Jacobian $[\partial f / \partial \mathbf{x}|_{\bar{\mathbf{x}}}]^T [\partial f / \partial \mathbf{x}|_{\bar{\mathbf{x}}}]$ for larger values of p . This increased stiffness is analogous to that 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 Sec. III that hyper-reduction can break this scaling, allowing for potentially inexpensive ROMs that only require f to be computed

for a few cells. In this section, we demonstrate LSPG, M-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 Fig. 10, which we hereafter refer to as sample mesh A and B. In sample mesh A the residual is sampled at 20,316 randomly selected cells (1.0% of all cells), requiring 445,373 cells (21.9% of all cells) to keep neighboring cells and the neighbors of neighbors needed for the FV stencil. Sample mesh B is a subset of sample mesh A, the residual is sampled at 16,253 randomly selected cells (0.8% of all cells), requiring 364,468 total cells (17.9% of all cells) required for each FV stencil.

Figure 11 shows the state error results. As for the full mesh ROM, CM-LSPG is more accurate than M-LSPG, with lower values of $\mathcal{E}_{\mathbf{x}}$ in almost all cases. CM-LSPG performs very well: although $\mathcal{E}_{\mathbf{x}}$ 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, because the

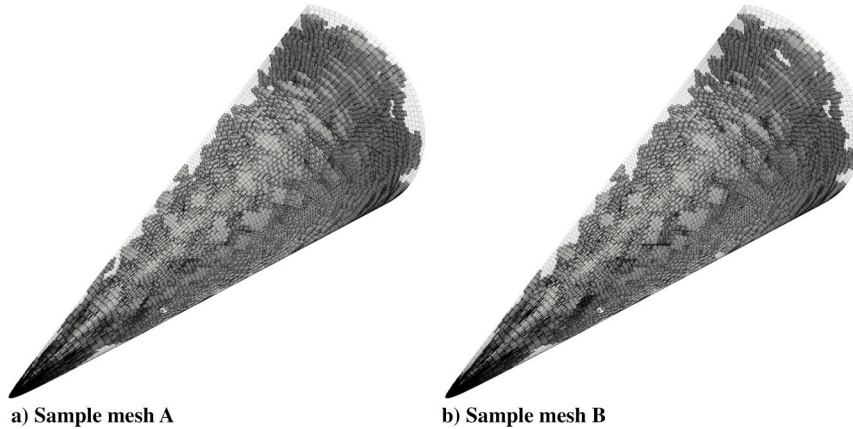


Fig. 10 HIFiRE-1 mesh with sample mesh cells highlighted.

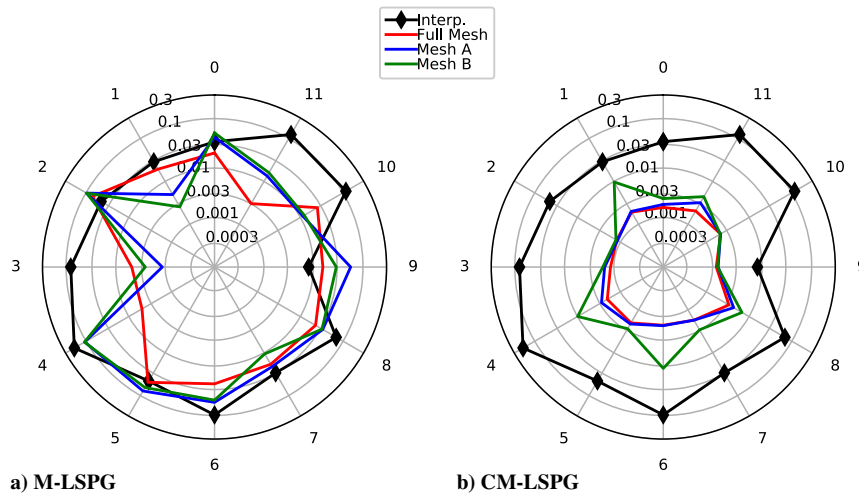


Fig. 11 Relative L^2 error \mathcal{E}_x computed on sample meshes with different ROM approaches. Case numbers correspond to the labels in Fig. 4.

Table 4 Case numbers for which the ROMs fail for different sample meshes. Case numbers correspond to the labels in Fig. 4

ROM type	Full mesh	Mesh A	Mesh B
LSPG	None	4	4
C-LSPG	None	None	2, 4, 6

clipper functions are active for three cases computed with sample mesh B. Note that the hyper-reduced CM-LSPG is only approximately conservative because we also compute the conservation constraint with the approximate residual $\mathbf{P}_r^T \mathbf{P}_r \mathbf{f}$ computed from the sample mesh.

Figure 12 shows the results for the wall heat flux, revealing that the hyper-reduced ROMs have mostly similar values the full mesh results. The ROMs clearly outperform the initial guess for only 5 of 12 test cases, and for most cases, $\mathcal{E}_{Q_{\text{wall}}}$ is smaller for CM-LSPG solutions than M-LSPG solutions. Additionally, the sample meshes tend to degrade the robustness of both LSPG and C-LSPG, as shown in Table 4. Less intuitive is the fact that unlike \mathcal{E}_x , $\mathcal{E}_{Q_{\text{wall}}}$ can be smaller for the sample mesh than the full mesh; see, for instance, cases 1 and 9 for sample mesh A. This is again due to the heat flux depending on the gradient of a derived quantity in a small region of the spatial domain; it seems that randomly selecting cells for the sample mesh can decrease or increase accuracy for quantities other than the state variables.

To complement the integrated heat flux errors $\mathcal{E}_{Q_{\text{wall}}}$ presented in this study, we present some results for the vehicle surface pressure

coefficient and heat flux fields. The pressure coefficient is shown because it is another important quantity of interest; it is used to compute aerodynamic forces and moments on a flight vehicle. Figure 13 shows the relative errors of the pressure coefficient for the initial guess and CM-LSPG solution of test case 10, computed on sample mesh B. We present results for test case 10 because it requires the ROM to extrapolate from the training set for a larger flight speed and freestream density. The results shown in Fig. 13 are similar to those obtained for the full mesh, just as $\mathcal{E}_{Q_{\text{wall}}}$ computed by CM-LSPG is similar for the full mesh and both sample meshes. The plots show that the CM-LSPG provides a roughly one order of magnitude decrease in error relative to the inverse distance interpolation used as an initial guess. The pressure coefficient error obtained from CM-LSPG is well under 1% on most of the vehicle surface. Note that the larger pressure coefficient errors present on the cylindrical midsection of the vehicle and just upstream of the flange correspond to a region in which the pressure coefficient is of low magnitude because it is changing from positive to negative. The heat flux error is slightly higher at around 1–3%. This is likely because the pressure coefficient depends on a derived field, as opposed to the gradient of a derived field.

For CM-LSPG, the \mathcal{E}_x shows that the smaller sample mesh (sample mesh B) is slightly less accurate in all cases except case 10. The impact of sample mesh for M-LSPG is less clear, likely due to the poor convergence of the residual shown in Fig. 14. Figure 14 also shows that the CM-LSPG residuals converge faster than the corresponding M-LSPG residuals, as observed for the full mesh. However, the convergence of the residual magnitude for both sample meshes is

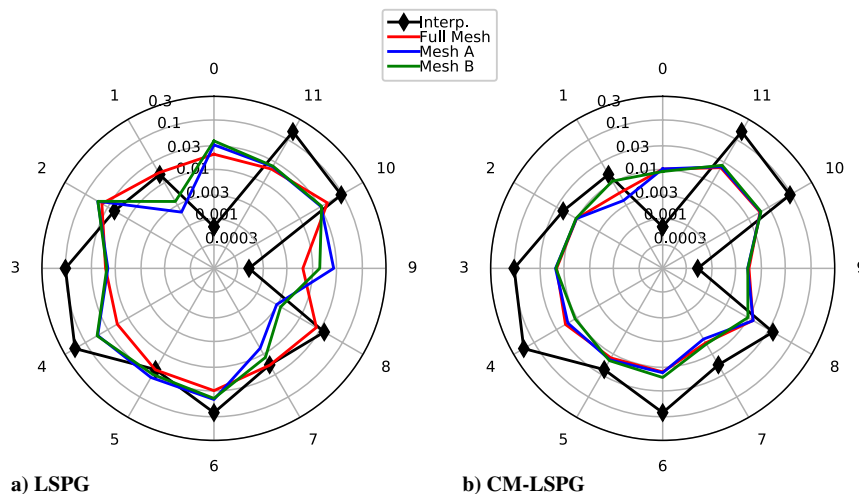


Fig. 12 Relative heat flux error $\mathcal{E}_{Q_{\text{wall}}}$ computed on sample meshes with different ROM approaches. Case numbers correspond to the labels in Fig. 4.

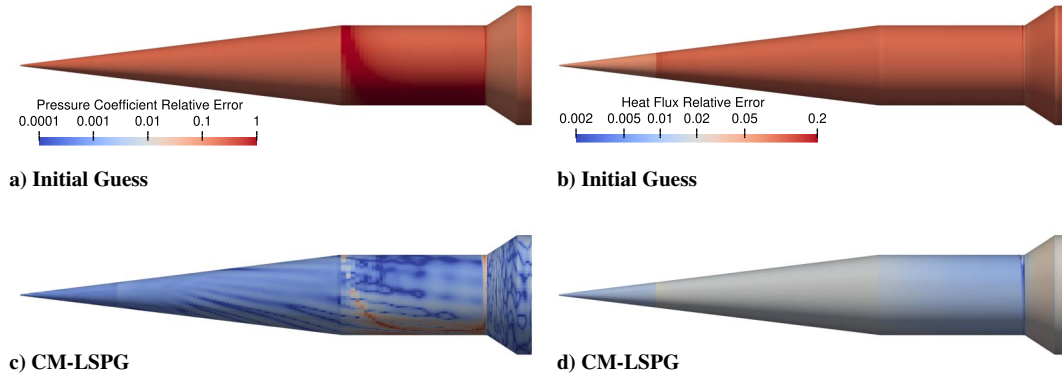


Fig. 13 Relative error fields for the CM-LSPG solution to test case 10, computed on sample mesh B.

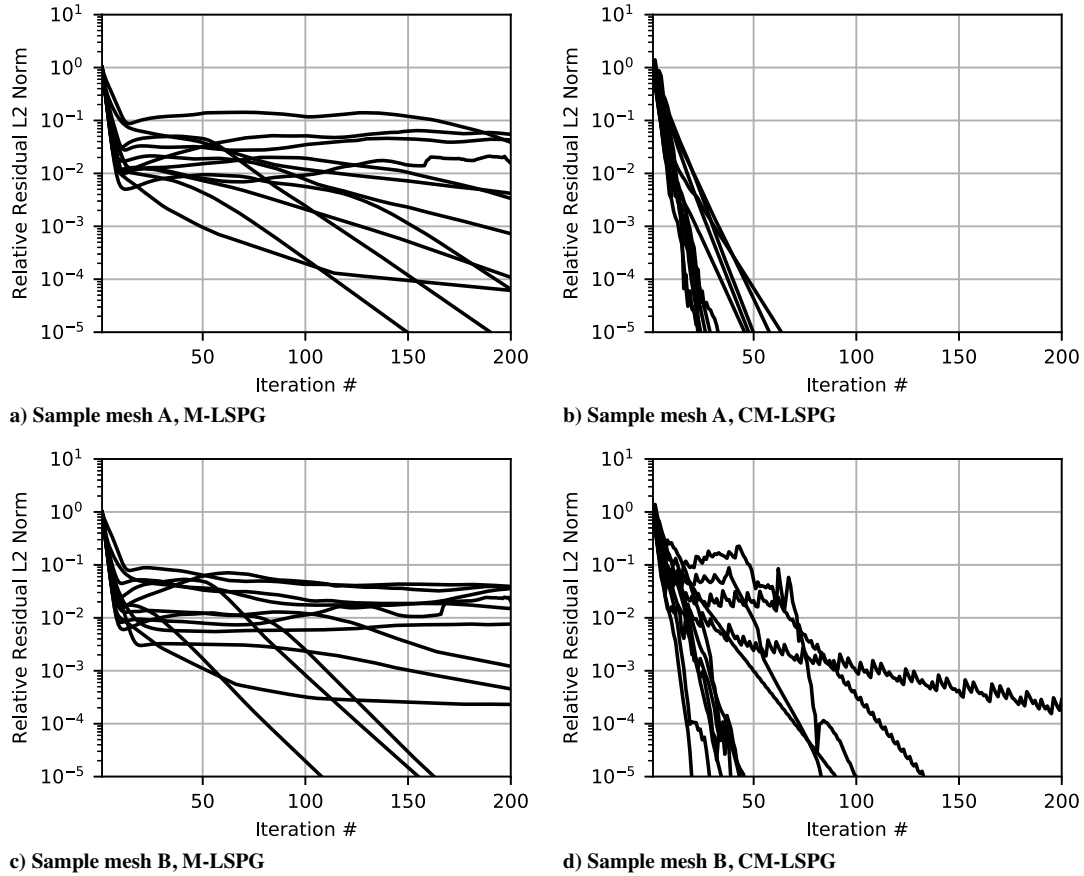


Fig. 14 Relative residual L2 norms for hyper-reduced M-LSPG and CM-LSPG computed with ROM dimension $p = 4$ on both sample meshes.

slower than that on the full mesh in some cases. In one case for sample mesh B, the residual fails to drop by five orders of magnitude in 200 iterations. Studies of coarser sample meshes not presented in this paper revealed that the CM-LSPG residual converges slower than the worst cases shown for sample mesh B.

In contrast to CM-LSPG, the convergence behavior of M-LSPG tends to improve on smaller sample meshes. This can be seen for three cases in Figs. 14a and 14c, and was also observed for smaller sample meshes whose results are not plotted for brevity. The convergence acceleration of M-LSPG on smaller meshes suggests that the convergence issues observed for 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 Eq. (10). 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 CM-LSPG on sample meshes.

3. ROM Performance

To estimate the performance of LSPG, M-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. The ROMs and FOMs were both run on 2.6 GHz Intel Sandy Bridge processors. The FOM cases were run with 128 cores, whereas the ROM cases were run on 16 cores, corresponding to a single cluster node. As in [15], the ROM speed-up estimates include the time needed to compute Q_{wall} as well as other wall quantities, e.g., pressure and friction coefficients, on a second postprocessing sample mesh that only contained the two layers of cells nearest to the wall needed to compute wall-normal gradients.

Figure 15 shows that the hyper-reduced ROMs are between 200 and 1000 times faster than the FOM for the test cases. The size of this

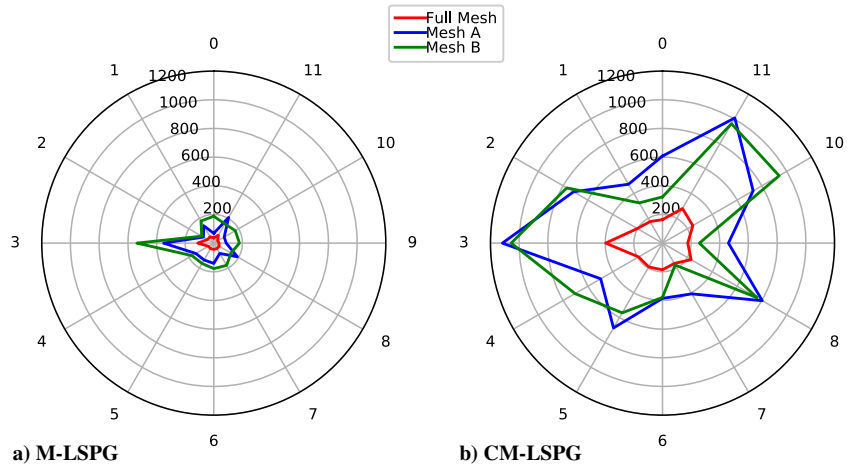


Fig. 15 Speed-up of the ROM over full-order model for LSPG and CM-LSPG. Case numbers correspond to the labels in Fig. 4.

speed-up means that hundreds or 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.

The performance of the steady ROM can be explained with a rudimentary complexity analysis of the FOM and ROM:

FOM : Solved with $\mathcal{O}(10,000)$ pseudo-time steps. Solved with one Newton step per pseudo-time step. Each Newton-step requires one residual and Jacobian evaluation with $\mathcal{O}(N)$ operations each. Additionally, $\mathcal{O}(100)$ Jacobian products are required for solving the linear system at each Newton step. Because the Jacobian is sparse, the cost of evaluating it and its product with a vector requires roughly $\mathcal{O}(3N)$ operations each, assuming a first-order Jacobian with roughly $3N$ nonzero entries. Therefore the cost of each FOM is around $\mathcal{O}(3,000,000N)$.

$$\text{Complexity of FOM} \sim \mathcal{O}(3,000,000N) \quad (24)$$

ROM : Most cases were solved with around $\mathcal{O}(30)$ Gauss–Newton iterations. Each Gauss–Newton step requires one (hyper-reduced) residual evaluation and Jacobian product. Each of these requires $\mathcal{O}(\nu N)$ operations, where ν is the fraction of cells in the sample mesh. For C/CM-LSPG, a $(p + n_u) \times (p + n_u)$ dense linear system is solved at each Gauss–Newton step. The n_u comes from the conservation constraint equations. The cost of solving an $n \times n$ dense linear system is $\mathcal{O}(n^3)$, and therefore a very rough cost estimate can be obtained from

$$\text{Complexity of CM-LSPG ROM} \sim \mathcal{O}(30\nu N(p + n_u)^3) \quad (25)$$

Firstly, much of the ROM speed-up comes from the much smaller number of nonlinear iterations; 3–4 orders of magnitude fewer than the FOM. This is the main reason for the speed-up obtained by the full mesh ROM over the FOM. Additional speed is obtained from the much smaller dimension of the LSPG minimization problem, $p \ll N$. For the case where $p = 4$ and $n_u = 6$, Eqs. (24) and (25) estimate that $\mathcal{O}(300)$ CM-LSPG ROMs can be run for the cost of one FOM, similar to the results shown in Fig. 15.

The further improvements in performance demonstrated by the hyper-reduced ROMs are made possible by the smaller cost of evaluating the hyper-reduced residual vector. In the implementation used for this study the residual is evaluated at each cell in the sample mesh. This makes the residual evaluation and Jacobian product roughly 4–6 times faster on the sample meshes considered. For the case where $p = 4$ and $\nu = 0.2$, Eqs. (24) and (25) estimate that $\mathcal{O}(500)$ hyper-reduced CM-LSPG ROMs can be run for the cost of one FOM, again similar to the results shown in Fig. 15.

It is important to point out that the ROM cost scales very poorly with the ROM dimension p . For the hyper-reduction factor $\nu = 0.2$ considered above, Eq. (25) estimates that a CM-LSPG ROM with

dimension $p = 70$ will cost roughly the same as an FOM. Because a fixed number of Gauss–Newton iterations is assumed in Eq. (25), and the number of iterations seems to increase with p for a given convergence tolerance, it is likely that the cost of the ROM with dimension $p < 70$ will exceed that of the FOM.

Fortunately, there is substantial room for improvement in the ROM performance presented in this paper. For instance, a more efficient ROM implementation in which the residual is only evaluated at sample points could decrease the ROM cost by an additional factor of 25 for a three-dimensional cell-centered FV mesh. Further improvements are possible with a more targeted hyper-reduction scheme such as GNAT. The low relative cost of the full mesh ROM means that the GNAT hyper-reduction technique [15], which requires residual snapshots taken from full mesh ROMs, may be feasible for steady hypersonic flow problems.

V. Conclusions

This work demonstrated the successful application of LSPG ROMs to steady-state flows around a hypersonic flight vehicle. This paper 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 FOM. The methods thus seem promising toward enabling many-query analyses for large steady-state hypersonic flow simulations.

The results of Sec. IV indicate that conservative manifold LSPG (CM-LSPG) is accurate and robust when trained with sufficient snapshot data over a parameter range. The two-parameter CM-LSPG ROM state errors are 0.1% and integrated heat 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 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, it was found that using a naive Gauss–Newton implementation with a linear basis can result in the solver finding nonphysical solutions with regions of negative temperature. These nonphysical states can be avoided by using a nonlinear basis that does not allow

for negative densities or temperatures. The increased robustness achieved by avoiding nonphysical 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, in order to increase the range of flight/design conditions over which a hypersonic ROM will be accurate and robust, a range of different basis choices should be explored. Some potential directions include affine POD bases with parameterized reference states $x^0(\mu)$, POD modes computed with physics-based inner-product definitions, a basis that can deform with shock movements, and other more sophisticated nonlinear state approximations such as autoencoders. Second, to apply CM-LSPG projection to models of flight vehicles that operate in flight regimes where nonequilibrium effects are important, appropriate clipping functions need to be derived for reacting gas models, and/or nonlinear mappings should be explored as an alternative to POD modes. Third, although collocation with random cell selection works well, there are likely better hyper-reduction strategies that can improve upon the ROM 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. Finally, the poor nonlinear convergence rates observed for some LSPG and CM-LSPG cases suggest that more work is needed on the LSPG and CM-LSPG nonlinear 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: Physical Conservation Laws

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

$$\frac{d}{dt} \int_{\omega} u_i(\mathbf{x}, t; \mu) d\mathbf{x} + \int_{\gamma} \mathbf{g}_i(\mathbf{x}, t; \mu) \cdot \mathbf{n}(\mathbf{x}) d\mathbf{s}(\mathbf{x}) = \int_{\omega} s_i(\mathbf{x}, t; \mu) d\mathbf{x},$$

$$i \in \mathbb{N}(n_u), \quad \forall \omega \subseteq \Omega \quad (\text{A1})$$

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 $u_i(\mathbf{x}, 0; \mu) = u_i^0(\mathbf{x}; \mu)$, $i \in \mathbb{N}(n_u)$, where $\mathbb{N}(a) := \{1, \dots, a\}$. Here, ω denotes any subset of the spatial domain of interest $\Omega \subset \mathbb{R}^d$ with $d \leq 3$; $\gamma := \partial\omega$ denotes the boundary of the subset ω , while $\Gamma := \partial\Omega$ denotes the boundary of the domain Ω ; $d\mathbf{s}(\mathbf{x})$ denotes integration with respect to the boundary; and $u_i \in \mathbb{R}$, $\mathbf{g}_i \in \mathbb{R}^d$, 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, $\mathbf{n} \in \mathbb{R}^d$ denotes the outward unit normal to ω . We emphasize that Eq. (A1) describes conservation of any set of variables u_i , $i \in \mathbb{N}(n_u)$, given their respective flux \mathbf{g}_i and source s_i functions.

Appendix B: Perfect Gas Reynolds-Averaged Navier–Stokes Equations

We consider the three-dimensional compressible Navier–Stokes equations with a turbulence model, which corresponds to Eq. (A1) with $d = 3$, and $n_u = 5, 6$, or 7 depending on the choice of turbulence model.^{††} The conserved quantities \mathbf{U} are defined in Eq. (1). For the case of $n_t = 0$, no turbulence equation is used and the 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., $\phi_1 = \tilde{v}$ 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

$$E = e + \frac{1}{2}(v_j v_j) \quad (\text{B1})$$

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

$$\begin{pmatrix} \mathbf{g}_1 \\ \vdots \\ \mathbf{g}_{n_u} \end{pmatrix} = \mathbf{F}_i(\mathbf{U}) - \mathbf{G}_i(\mathbf{U}) \quad (\text{B2})$$

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

$$\mathbf{F}_i(\mathbf{U}) = \begin{pmatrix} \rho v_i \\ \rho v_i v_j + P \delta_{ij} \\ \rho E v_i + P v_i \\ F_{i,1}^{\text{turb}} \\ \vdots \\ F_{i,n_t}^{\text{turb}} \end{pmatrix} \quad (\text{B3})$$

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

$$\mathbf{G}_i(\mathbf{U}) = \begin{pmatrix} 0 \\ \tau_{ij} \\ \tau_{ij} v_j - q_i \\ G_{i,1}^{\text{turb}} \\ \vdots \\ G_{i,n_t}^{\text{turb}} \end{pmatrix} \quad (\text{B4})$$

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, $G_{i,t}^{\text{turb}}$ 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 that account solely for advection. Because 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 that relates the viscosity and spatial derivatives of the velocity to the stresses. For a Newtonian fluid (i.e., one that has a linear stress–strain relationship), the deviatoric stress tensor is often written as

$$\tau_{ij} = \mu \left(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i} \right) + \lambda \delta_{ij} \left(\frac{\partial v_k}{\partial x_k} \right) \quad (\text{B5})$$

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:

^{††}Note that additional conservation equations are required for nonperfect gases, and nonequilibrium chemistry, but these will not be considered in this paper.

$$q_i = -\kappa \frac{\partial T}{\partial x_i} \quad (\text{B6})$$

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

Lastly the source vector \mathbf{S} is written as

$$\mathbf{S}(\mathbf{U}) \equiv \begin{pmatrix} s_1 \\ \vdots \\ s_{n_u} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ S_1 \\ \vdots \\ S_{n_t} \end{pmatrix} \quad (\text{B7})$$

where S_i 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

$$P = \rho RT \quad (\text{B8})$$

where R is a constant specific to the type of gas (for air $R = 287.1 \text{ J}/(\text{kg} \cdot \text{K})$). The calorically perfect gas assumption has the following requirements: 1) the gas is in thermal equilibrium, 2) the gas is not chemically reacting, 3) the internal energy and enthalpy are dependent only on temperature, and 4) 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

$$e = c_v T, \quad h = c_p T \quad (\text{B9})$$

and the specific heats are written as

$$c_v = \frac{R}{\gamma - 1}, \quad c_p = \frac{\gamma R}{\gamma - 1} \quad (\text{B10})$$

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

$$\gamma = \frac{c_p}{c_v} \quad (\text{B11})$$

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. (B8), we obtain the following form of the ideal gas equation:

$$P = (\gamma - 1)\rho e \quad (\text{B12})$$

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

$$M = \frac{\sqrt{v_j v_j}}{C} \quad (\text{B13})$$

and the speed of sound is computed by

$$C = \sqrt{\gamma \frac{P}{\rho}} \quad (\text{B14})$$

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

$$\mu = \mu_{\text{ref}} \frac{T^{3/2}}{T + T_{\text{ref}}} \quad (\text{B15})$$

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

$$\mu = \mu_{\text{ref}} \left(\frac{T}{T_{\text{ref}}} \right)^{3/2} \frac{T_{\text{ref}} + S}{T + S} \quad (\text{B16})$$

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

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

$$\kappa = \frac{c_p \mu}{Pr} \quad (\text{B17})$$

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 Eq. (A1) are completed by a set of boundary conditions

$$\mathbf{b}(\mathbf{U}) = \bar{\mathbf{b}}(\mathbf{U}, \mathbf{x}; \boldsymbol{\mu}) \quad \text{on} \quad \Gamma_f \quad (\text{B18})$$

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

$$\mathbf{F}_i(\mathbf{U}) = \bar{\mathbf{F}}_i(\mathbf{U}, \mathbf{x}; \boldsymbol{\mu}) \quad \text{on} \quad \Gamma_{\bar{f}} \quad (\text{B19})$$

and

$$\mathbf{G}_i(\mathbf{U}) = \bar{\mathbf{G}}_i(\mathbf{U}, \mathbf{x}; \boldsymbol{\mu}) \quad \text{on} \quad \Gamma_{\bar{f}} \quad (\text{B20})$$

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

$$\mathbf{U}(\mathbf{x}, t = 0; \boldsymbol{\mu}) = \mathbf{U}^0(\mathbf{x}; \boldsymbol{\mu}) \quad \text{in} \quad \Omega_f \quad (\text{B21})$$

Appendix C: Finite-Volume Discretization

To discretize the governing Eq. (A1), we apply the finite-volume method [20,21], as it explicitly enforces conservation over prescribed control volumes. In particular, we assume that the spatial domain $\Omega \subset \mathbb{R}^d$ with $d \leq 3$ has been partitioned into a mesh \mathcal{M} , of $N_\Omega \in \mathbb{N}$ nonoverlapping (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 $\Gamma_i = \{\mathbf{x} | \mathbf{x} \in e, \forall e \in \mathcal{E}_i, e \in \mathbb{N}(|\mathcal{E}_i|)\}$. Then the full set of N_e faces within the mesh is $\mathcal{E} \equiv \{e_i\}_{i=1}^{N_e} := \bigcup_{i=1}^{N_\Omega} \mathcal{E}_i$. Enforcing conservation (A1) on each control volume in the mesh yields

$$\begin{aligned} \frac{d}{dt} \int_{\Omega_j} u_i(\mathbf{x}, t; \boldsymbol{\mu}) d\mathbf{x} + \int_{\Gamma_j} \mathbf{g}_i(\mathbf{x}, t; \boldsymbol{\mu}) \cdot \mathbf{n}_j(\mathbf{x}) d\mathbf{s}(\mathbf{x}) \\ = \int_{\Omega_j} s_i(\mathbf{x}, t; \boldsymbol{\mu}) d\mathbf{x}, \quad i \in \mathbb{N}(n_u), j \in \mathbb{N}(N_{\Omega}) \end{aligned} \quad (C1)$$

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

$$x_{\mathcal{I}(i,j)}(t; \boldsymbol{\mu}) = \frac{1}{|\Omega_j|} \int_{\Omega_j} u_i(\mathbf{x}, t; \boldsymbol{\mu}) d\mathbf{x}, \quad i \in \mathbb{N}(n_u), j \in \mathbb{N}(N_{\Omega}) \quad (C2)$$

where $\mathcal{I}: \mathbb{N}(n_u) \times \mathbb{N}(N_{\Omega}) \rightarrow \mathbb{N}(N)$ denotes a mapping from conservation-law index and control-volume index to degree of freedom, and a velocity vector $\mathbf{f}(\mathbf{w}, \tau; \boldsymbol{\nu}) = \mathbf{f}^g(\mathbf{w}, \tau; \boldsymbol{\nu}) + \mathbf{f}^s(\mathbf{w}, \tau; \boldsymbol{\nu})$ with $\mathbf{f}^g, \mathbf{f}^s \in \mathbb{R}^N$ whose elements consist of

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

for $i \in \mathbb{N}(n_u), j \in \mathbb{N}(N_{\Omega})$. Here, the fields $\mathbf{g}_i^{\text{FV}} \in \mathbb{R}^d$ and $s_i^{\text{FV}} \in \mathbb{R}$, $i \in \mathbb{N}(n_u)$, denote the approximated flux and source, respectively, associated with the i th conserved variable (per unit area per unit time). Substituting $\int_{\Omega_j} u_i(\mathbf{x}, t; \boldsymbol{\mu}) d\mathbf{x} \leftarrow |\Omega_j| x_{\mathcal{I}(i,j)}(t; \boldsymbol{\mu})$, $\mathbf{g}_i \leftarrow \mathbf{g}_i^{\text{FV}}$, and $s_i \leftarrow s_i^{\text{FV}}$ in Eq. (C1) and dividing by $|\Omega_j|$ yields

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, t; \boldsymbol{\mu}), \quad \mathbf{x}(0; \boldsymbol{\mu}) = \mathbf{x}^0(\boldsymbol{\mu}) \quad (C3)$$

where

$$x_{\mathcal{I}(i,j)}^0(\boldsymbol{\mu}) := \frac{1}{|\Omega_j|} \int_{\Omega_j} u_i^0(\mathbf{x}; \boldsymbol{\mu}) d\mathbf{x}$$

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 FOM. We hereafter refer to Eq. (C3) as the FOM ODE.

In the case of a steady-state solution, the velocity exhibits no time dependence such that the FOM ODE (C3) reduces to

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

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

Appendix D: 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 $\bar{\Omega}_i = \bigcup_{j \in \mathcal{K} \subseteq \mathbb{N}(N_{\Omega})} \Omega_j$, $i \in \mathbb{N}(N_{\bar{\Omega}})$ 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 $\bar{\Gamma}_i = \{\mathbf{x} | \mathbf{x} \in e, \forall e \in \bar{\mathcal{E}}_i, \bar{\mathcal{E}}_i \in \mathbb{N}(|\bar{\mathcal{E}}|)\} \subseteq \bigcup_{j=1}^{N_{\bar{\Omega}}} \Gamma_j$, $i \in \mathbb{N}(N_{\bar{\Omega}})$ 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 $\bar{\mathcal{E}} := \bigcup_{i=1}^{N_{\bar{\Omega}}} \bar{\mathcal{E}}_i \subseteq \mathcal{E}$. 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 (A1) on each subdomain in the decomposed mesh yields

$$\begin{aligned} \int_{\bar{\Gamma}_j} \mathbf{g}_i(\mathbf{x}, t; \boldsymbol{\mu}) \cdot \bar{\mathbf{n}}_j(\mathbf{x}) d\mathbf{s}(\mathbf{x}) &= \int_{\bar{\Omega}_j} s_i(\mathbf{x}, t; \boldsymbol{\mu}) d\mathbf{x}, \\ i \in \mathbb{N}(n_u), j \in \mathbb{N}(N_{\bar{\Omega}}) \end{aligned} \quad (D1)$$

where $\bar{\mathbf{n}}_j \in \mathbb{R}^d$ denotes the unit normal to subdomain $\bar{\Omega}_j$. We propose applying the same finite-volume discretization employed to discretize the control-volume conservation Eq. (C1) to the subdomain conservation Eq. (D1). To accomplish this, we introduce a decomposed state vector $\bar{\mathbf{x}} \in \mathbb{R}^{\bar{N}}$ with $\bar{N} = N_{\bar{\Omega}} n_u$ and elements

$$\bar{x}_{\bar{\mathcal{I}}(i,j)}(\mathbf{x}; \boldsymbol{\mu}) = \frac{1}{|\bar{\Omega}_j|} \int_{\bar{\Omega}_j} u_i(\mathbf{x}; \boldsymbol{\mu}) d\mathbf{x}, \quad i \in \mathbb{N}(n_u), j \in \mathbb{N}(N_{\bar{\Omega}}) \quad (D2)$$

where $\bar{\mathcal{I}}: \mathbb{N}(n_u) \times \mathbb{N}(N_{\bar{\Omega}}) \rightarrow \mathbb{N}(\bar{N})$ 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 \mathbf{x} as

$$\bar{\mathbf{x}}(\mathbf{x}) = \bar{\mathbf{C}} \mathbf{x}$$

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

$$\bar{C}_{\bar{\mathcal{I}}(i,j), \mathcal{I}(l,k)} = \frac{|\Omega_k|}{|\bar{\Omega}_j|} \delta_{il} I(\Omega_k \subseteq \bar{\Omega}_j) \quad (D3)$$

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

$$\bar{\mathbf{f}}(\mathbf{x}; \boldsymbol{\mu}) = \bar{\mathbf{C}} \mathbf{f}(\mathbf{x}; \boldsymbol{\mu}) \quad (D4)$$

such that subdomain conservation can be expressed as

$$\bar{\mathbf{C}} \mathbf{f}(\mathbf{x}; \boldsymbol{\mu}) = \mathbf{0} \quad (D5)$$

For a more detailed explanation on the derivation of Eqs. (D4) and (D5), we refer readers to Ref. [18] (Sec. 4.1).

Acknowledgments

This study was supported by the Laboratory Directed Research and Development program at Sandia National Laboratories, a multi-mission 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. 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.

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>
- [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>
- [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>
- [4] Dutta, S., Braun, R. D., and Karlgaard, C. D., "Uncertainty Quantification for Mars Entry, Descent, and Landing Reconstruction Using Adap-

- tive Filtering,” *Journal of Spacecraft and Rockets*, Vol. 51, No. 3, 2014, pp. 967–977.
<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 Paper 2010-7930, 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. 1625–1646.
<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. 74–84.
<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. 994–1009.
<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. 2358–2368.
<https://doi.org/10.2514/1.J055499>
 - [10] Abgrall, R., and Crisovan, R., “Model Reduction Using L^1 -Norm Minimization as an Application to Nonlinear Hyperbolic Problems,” *International Journal for Numerical Methods in Fluids*, Vol. 87, No. 12, 2018, pp. 628–651.
<https://doi.org/10.1002/flid.4507>
 - [11] Lucia, D. J., King, P. I., and Beran, P. S., “Reduced Order Modeling of a Two-Dimensional Flow with Moving Shocks,” *Computers and Fluids*, Vol. 32, No. 7, 2003, pp. 917–938.
[https://doi.org/10.1016/S0045-7930\(02\)00035-X](https://doi.org/10.1016/S0045-7930(02)00035-X)
 - [12] Abgrall, R., Amsallem, D., and Crisovan, R., “Robust Model Reduction by L^1 -Norm Minimization and Approximation via Dictionaries: Application to Nonlinear Hyperbolic Problems,” *Advanced Modeling and Simulation in Engineering Sciences*, Vol. 3, No. 1, 2016, p. 1.
<https://doi.org/10.1186/s40323-015-0055-3>
 - [13] Nair, N. J., and Balajewicz, M., “Transported Snapshot Model Order Reduction Approach for Parametric, Steady-State Fluid Flows Containing Parameter-Dependent Shocks,” *International Journal for Numerical Methods in Engineering*, Vol. 117, No. 12, 2019, pp. 1234–1262.
<https://doi.org/10.1002/nme.5998>
 - [14] Carlberg, K., 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.
<https://doi.org/10.1002/nme.3050>
 - [15] 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.
<https://doi.org/10.1016/j.jcp.2013.02.028>
 - [16] Washabaugh, K. M., “Fast Fidelity for Better Design: A Scalable Model Order Reduction Framework for Steady Aerodynamic Design Applications,” Ph.D. Dissertation, Dept. of Aeronautics and Astronautics, Stanford Univ., Stanford, CA, Aug. 2016.
 - [17] Carlberg, K., Barone, M., and Antil, H., “Galerkin v. Least-Squares Petrov-Galerkin Projection in Nonlinear Model Reduction,” *Journal of Computational Physics*, Vol. 330, Feb. 2017, pp. 693–734.
<https://doi.org/10.1016/j.jcp.2016.10.033>
 - [18] Carlberg, K., Choi, Y., and Sargsyan, S., “Conservative Model Reduction for Finite-Volume Models,” *Journal of Computational Physics*, Vol. 371, Oct. 2018, pp. 280–314.
<https://doi.org/10.1016/j.jcp.2018.05.019>
 - [19] 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*, AIAA Paper 2017-4407, 2017.
<https://doi.org/10.2514/6.2017-4407>
 - [20] LeVeque, R. J., *Finite Volume Methods for Hyperbolic Problems*, Vol. 31, Cambridge Univ. Press, Cambridge, England, U.K., 2002, Chap. 4.
<https://doi.org/10.1017/CBO9780511791253>
 - [21] Eymard, R., Gallouët, T., and Herbin, R., “Finite Volume Methods,” *Handbook of Numerical Analysis*, Vol. 7, 2000, pp. 713–1018.
[https://doi.org/10.1016/S1570-8659\(00\)07005-8](https://doi.org/10.1016/S1570-8659(00)07005-8)
 - [22] Holmes, P., Lumley, J., and Berkooz, G., *Turbulence, Coherent Structures, Dynamical Systems and Symmetry*, Cambridge Univ. Press, Cambridge, England, U.K., 1996, Chap. 3.
<https://doi.org/10.1017/CBO9780511622700>
 - [23] Prud’homme, C., Rovas, D. V., Veroy, K., Machiels, L., Maday, Y., Patera, A. T., and Turinici, G., “Reliable Real-Time Solution of Parametrized Partial Differential Equations: Reduced-Basis Output Bound Methods,” *Journal of Fluids Engineering*, Vol. 124, No. 1, 2001, pp. 70–80.
<https://doi.org/10.1115/1.1448332>
 - [24] 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.
<https://doi.org/10.1007/BF03024948>
 - [25] Huang, C., Duraisamy, K., and Merkle, C., “Investigations and Improvement of Robustness of Reduced-Order Models of Reacting Flow,” *AIAA SciTech 2019 Forum*, AIAA Paper 2019-2012, 2019.
<https://doi.org/10.2514/6.2019-2012>
 - [26] Lee, K., and Carlberg, K. T., “Model Reduction of Dynamical Systems on Nonlinear Manifolds Using Deep Convolutional Autoencoders,” *Journal of Computational Physics*, Vol. 404, March 2020, Paper 108973.
<https://doi.org/10.1016/j.jcp.2019.108973>
 - [27] 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>
 - [28] 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,” Sandia National Lab. Rept. SAND2019-10842, Albuquerque, NM, 2019.
 - [29] Lee, K., Carlberg, K., and Elman, H. C., “Stochastic Least-Squares Petrov-Galerkin Method for Parameterized Linear Systems,” *SIAM/ASA Journal on Uncertainty Quantification*, Vol. 6, No. 1, 2018, pp. 374–396.
<https://doi.org/10.1137/17M1110729>

K. Taira
Associate Editor