

Reduced-Order Modeling of an Entry Capsule from Simulated Ballistic Range Trajectories

Hayden V. Dean,^{*} Christian Perron,[†] Bradford E. Robertson,[‡] and Dimitri N. Mavris[§]
Georgia Institute of Technology, Atlanta, Georgia 30332

<https://doi.org/10.2514/1.A36396>

Entry, descent, and landing (EDL) relies on blunt-body entry vehicles, or entry capsules, to survive atmospheric entry and successfully complete planetary missions. In the supersonic through the transonic flight regime, these entry capsules can display dynamic instabilities. While research is ongoing to leverage nonlinear functional fits to determine dynamic stability coefficients, advancements in computational fluid dynamics–rigid body dynamics (CFD-RBD) simulations produce rich aerodynamic data, which might aid researchers in understanding and quantifying the dynamic stability characteristics of these capsules. The objective of this research is to build upon previous work that developed an aerodynamic Reduced-Order Model (ROM) to predict forced rotational oscillation trajectories, addressing key limitations. A ROM is a form of field-based surrogate model capable of efficiently leveraging high-fidelity aerodynamic data to mimic the results of CFD simulations. Results showed success at predicting the mean aerodynamic trends of a free-flight validation trajectory for C_P and C_F data for both the heatshield and backshell regions using a ROM methodology.

Nomenclature

A	= batch snapshot matrix
a_{ref}	= speed of sound
b	= batch number of snapshots
c	= number of inducing points
C_A	= axial force coefficient
C_F	= skin friction coefficient
C_l	= rolling moment coefficient
C_m	= pitching moment coefficient
$C_{m,\alpha}$	= C_m as a function of α
C_N	= normal force coefficient
C_n	= yaw moment coefficient
C_P	= coefficient of pressure
C_Y	= side force coefficient
D	= singular value matrix
$\tilde{D}$	= batched singular value matrix
d	= ROM dimension
ff	= forget factor of streaming singular value decomposition
f_n	= natural frequency
g	= regression function operator
h	= preserved left singular vectors
I	= identity matrix
I_{yy}	= moment of inertia in y-direction
i, j	= generic indices
K	= generic covariance matrix
K_{cc}	= covariance matrix between inducing points
K_{mm}	= exact covariance matrix

K_{mc}	= covariance matrix between input samples and inducing points
$k(\cdot)$	= kernel function
L_{ref}	= reference length
M	= Mach number
m	= number of snapshots
n	= degrees of freedom of samples
p	= vector of design variables
Q	= orthogonal basis of incorporated data
Q_{mm}	= sparse approximation of covariance matrix
$\hat{q}$	= dynamic pressure
R	= upper triangular matrix of incorporated data
S	= data scatter matrix
S_{ref}	= cross-sectional area
U	= left singular vectors of R or mode matrix
U'	= partial mode matrix
$\tilde{U}$	= batched partial mode matrix
$\hat{U}$	= truncated partial mode matrix
V	= right singular vectors of R
$\tilde{V}$	= batched right singular vectors R_i
v	= number of validation snapshots
w_j	= j th weight of regression model
X	= matrix of snapshots
x	= single snapshot
$\bar{x}$	= snapshot matrix mean
Z	= matrix of latent coordinates
z	= single latent coordinate
z_i	= i th coordinate of z
α	= angle of attack
α_{total}	= total angle of attack
$\dot{\alpha}_{\text{total}}$	= total rate of angle of attack
β	= side slip angle
Δ	= normalized distance function
δ	= desired information content
ϵ	= observed noise
θ	= pitch angle
$\dot{\theta}$	= pitch rate
θ_k	= length scale of covariance function
μ	= mean of prior Gaussian distribution
ρ	= atmospheric density
Σ	= singular value matrix
σ	= singular value
σ_{ij}^2	= variance of covariance function
σ_n^2	= noise variance
Φ	= POD mode matrix
ϕ	= POD mode

Presented as Paper 2025-2010 at the AIAA SciTech Forum, Orlando, FL, January 6–10, 2025; received 24 February 2025; accepted for publication 11 August 2025; published online Open Access 31 December 2025. Copyright © 2025 by Hayden V. Dean, Christian Perron, Bradford E. Robertson, and Dimitri N. Mavris. Published by the American Institute of Aeronautics and Astronautics, Inc., with permission. All requests for copying and permission to reprint should be submitted to CCC at www.copyright.com; employ the eISSN 1533-6794 to initiate your request. See also AIAA Rights and Permissions <https://aiaa.org/publications/publish-with-aiaa/rights-and-permissions/>.

^{*}Graduate Researcher, Aerospace Systems Design Laboratory (ASDL), School of Aerospace Engineering, Student Member AIAA.

[†]Research Engineer II, ASDL, School of Aerospace Engineering, Member AIAA.

[‡]Senior Research Engineer, ASDL, School of Aerospace Engineering; bradford.robertson@asdl.gatech.edu. Senior Member AIAA.

[§]S. P. Langley Distinguished Regents Professor and Director of ASDL, Fellow AIAA.

ϕ	=	roll angle
$\dot{\phi}$	=	roll rate
ψ	=	yaw angle
$\dot{\psi}$	=	yaw rate
ω_n	=	natural angular frequency

I. Introduction

THE entry, descent, and landing (EDL) for planetary missions have utilized blunt-body vehicles, or entry capsules, for several decades. Entry capsules are designed to withstand the stress of entry, effectively dissipate heat, and slow the vehicle down to safely land scientific payloads onto a planet's surface. However, a critical challenge with entry capsules is that they display dynamic instabilities in the supersonic to transonic flight regimes, which is exacerbated by the vehicle's rapid deceleration [1]. Accurately quantifying this dynamic stability behavior both through physical testing methods [1–4] and computational fluid dynamics (CFD) simulations [5–11] has been a significant research effort. Current practices of dynamic stability quantification make linear and quasi-static aerodynamic assumptions, utilizing assumed functional forms to derive dynamic stability coefficients from trajectory-fitting techniques [1]. Research is ongoing to account for the nonlinear dynamic nature of these vehicles with nonlinear functional forms for dynamic stability coefficients [12,13]. Research by Ernst et al. [7] avoided dynamic stability coefficients altogether by using neural networks to predict time-accurate forces and moments on the vehicle. These research efforts highlight that the focus of recent dynamic stability quantification is devoted to finding more physically consistent functional forms to represent the dynamic aerodynamic characteristics of entry vehicles. Since the aerodynamic response of the vehicle is driven by surface pressure and skin friction fields [14], what if there were a way to efficiently predict these aerodynamic sources? This would capture the vehicle behavior and eliminate the need for linear and quasi-static dynamic stability coefficients [15].

Reduced-order models (ROMs) are a type of surrogate model with the ability to efficiently leverage high-fidelity aerodynamic data to predict fields of data [15–22]. ROMs mimic the results of CFD simulations by predicting the full aerodynamic field of interest [15,22]. ROMs find a low-dimensional representation, or *latent space*, of the high-fidelity aerodynamic data, capturing the dominant and fundamental characteristics of the original data [15,16,23]. By utilizing this latent space, predictions of the unsampled high-dimensional space can be generated, producing similar results as CFD solutions but with less online computational cost. ROMs are actively being applied to fluid flow problems similar to blunt-body entry vehicles, such as vortex shedding [24], turbulent flows [25,26], compressible flow [17], unsteady aerodynamics [27], and store separation [28].

Dean et al. [15] previously applied a ROM methodology to predict pressure distributions on the surface of the Genesis Sample Return Capsule (SRC) ballistic range model resulting from forced pitching oscillation trajectories. The results showed success in predicting an unsampled trajectory, with greater success occurring with the heatshield predictions than with the backshell. The authors outlined some limitations of the work, which are summarized here. Firstly, the CFD grid was only suitable for resolving pressure forces; therefore, skin friction predictions were not explored and may be critical to understanding dynamic instability. Secondly, some grid refinement was needed to better resolve the flow over the shoulder region of the entry capsule, as most of the error in the heatshield predictions was related to this region. Thirdly, the ROMs were constructed using kriging regression, which is known to scale poorly with large amounts of training data; using approximate kriging methods was suggested to efficiently incorporate the large amount of data that the CFD produces. Finally, Dean et al. [15] stated that the ROMs were constructed from forced rotational oscillation data which could result in the vehicle having an unrealistic behavior when used in free-flight trajectory prediction. Applying the ROM methodology to free-flight CFD data would progress the methodology further by capturing more physically consistent dynamics [15].

This research advances the work of Dean et al. [15] by updating the ROM methodology to account for more representative CFD-in-the-loop flight simulation data to produce both pressure and skin friction ROMs. Additionally, this research contributes to the quantification of the dynamic stability of blunt-body entry vehicles by creating an aerodynamic ROM capable of predicting aerodynamic forces and moments without traditional linearized aerodynamic assumptions.

II. Reduced-Order Modeling Methodology

The ROMs outlined in this research are classified as nonintrusive, or data-driven, requiring only the inputs and outputs to be trained [16,29]. This is favorable in many applications involving *black-box* analysis codes, like in cases leveraging industry-standard CFD simulations [15,16,29]. When it comes to fluid flow analysis with ROMs, applications are divided into two categories: parametrically varying fluid flows and time-varying fluid flows [15]. Parametrically varying ROMs are built on the parametric variation (i.e., variations in Mach number M or angle of attack α) of steady flow-fields [15,30]. In contrast, time-varying ROMs are built on modal decomposition and identifying underlying structures in unsteady fluid flows [15]. To build an entry vehicle ROM that captures the dynamic response of the vehicle resulting from the flowfield and the vehicle state, it is critical to develop a parametric and time-varying ROM [15].

As stated by Dean et al. [15], a ROM can be divided into four main components: training data generation, dimensionality reduction (DR), interpolation or regression, and backmapping. The ROM methodology outlined in [15] is summarized below, with several key additions for this application.

A. Training Data Generation

This research focuses on developing an entry capsule ROM capable of predicting the pressure and skin friction distributions over the surface of a vehicle. The pressure and shear stress training data are generated using a CFD-in-the-loop flight simulation environment that combines the six-degree-of-freedom (6DOF) trajectory simulation tool, Program to Optimize Simulated Trajectories II (POST2), and the CFD simulation tool, Fully Unstructured Navier–Stokes (FUN3D). This tool was first developed by Ernst et al. [31] and has been applied to the Army–Navy Finner (ANF) projectile, Supersonic Inflatable Aerodynamic Decelerator (SIAD) ballistic range tests [31–33], Mars Science Laboratory (MSL) Reaction Control System (RCS) jet firing [5], and high-fidelity human-scale Mars Lander simulations [34,35]. Utilizing the POST2/FUN3D framework, the vehicle can be simulated in full 6DOF motion, providing time-accurate aerodynamics and dynamics of the vehicle.

To construct the ROM, the aerodynamic fields of interest must be related to all relevant vehicle states. Ideally, relevant state variables would be selected and variable ranges would be established for the entire trajectory space. Then, a design of experiments (DoE) would be generated to capture the vehicle state space most efficiently. The samples of such a DoE would normally be uncorrelated to avoid introducing any bias to the model. Finally, the simulations would be run for each sample of the DoE. However, this procedure is not possible with free-flight simulations, as the vehicle states are only explicitly prescribed at the initial conditions. Therefore, the following is a key research question: How can CFD-in-the-loop flight simulation be sampled to populate the free-flight ROM input space?

As outlined in [36], only the initial configuration of the vehicle at release can be controlled, with the vehicle motion resulting from aerodynamic interactions. Each point along the flown trajectory provides a potential training point for the ROM, but the vehicle states captured by a single trajectory are unknown until it is flown. With multiple flown trajectories, the observed vehicle states are unlikely to be uniformly distributed, i.e., the trajectories could converge, diverge, or remain constant throughout the state space. Similarly, gaps in the state space sampling are possible, as those regions could correspond to states that a vehicle would rarely encounter in real flight. Such nonideal sampling distribution will

affect the quality of the ROM and potentially result in poor predictions. To combat this issue, the following sampling methodology was implemented:

1) Identify the vehicle state variables of interest and their relevant bounds.

2) Apply an initial DoE sampling to the first, or highest, M_i , varying all other defined state parameters.

3) At each initial condition, simulate the free-flight trajectory, terminating the run at the final M_f or lowest M that the vehicle decelerates to, which has been established earlier.

By following this sampling methodology, the free-flight state space should be sufficiently captured, enabling the generation of a free-flight ROM. Specifics about how the sampling was conducted for the given application are discussed in later sections.

B. Dimensionality Reduction

Dimensionality reduction (DR) is applied to the generated training data to establish the low-dimensional space for the ROM. Previous research has leveraged principal component analysis (PCA) or proper orthogonal decomposition (POD) to effectively reduce aerodynamic surface distributions resulting from the forced rotational oscillation of the Genesis SRC ballistic range model [15]. POD is an unsupervised, linear DR method that utilizes eigenvalue decomposition to extract the main features of the data [37]. The objective of POD is to decompose the full-order model (FOM) into basis functions called *modes* that capture the highest amount of energy or variance in the data in the least number of modes necessary [37]. The following section summarizes the methodology.

From the free-flight simulations, a snapshot matrix $\mathbf{X} = [\mathbf{x}^1, \dots, \mathbf{x}^m] \in \mathbb{R}^{n \times m}$ is generated from m snapshots $\mathbf{x}$ having n degrees of freedom. For this ROM application, a snapshot $\mathbf{x}$ is a sample of the aerodynamic field of interest, m represents the total number of samples produced from all the simulated free-flight trajectories, and n represents the size of each field. Specifically, the pressure and wall shear stress fields, or C_p and C_f in nondimensional form, are calculated at the CFD grid surface points. Note that the wall shear stress is a vector field, meaning that each grid point holds three shear stress values and that the resulting snapshot is three times the size of the grid. The input matrix $\mathbf{X}$ is mean-centered before applying POD, resulting in [15,16]

$$\bar{\mathbf{x}} = \frac{1}{m} \sum_j \mathbf{x}_j = 0 \quad (1)$$

After centering, the high-dimensional snapshot matrix $\mathbf{X}$ is projected into the low-dimensional latent space composed of the POD modes, represented as orthonormal basis vectors $\boldsymbol{\phi}$ [15,16]. These are collected into a mode matrix $\boldsymbol{\Phi} = [\boldsymbol{\phi}^1, \dots, \boldsymbol{\phi}^d] \in \mathbb{R}^{n \times d}$, where the number of selected modes d defines the dimension of the latent space [15,16]. The POD modes are defined such that the following Rayleigh quotient is maximized [38]:

$$\max_{\boldsymbol{\phi}_j} \frac{\boldsymbol{\phi}_j^T \mathbf{S} \boldsymbol{\phi}_j}{\boldsymbol{\phi}_j^T \boldsymbol{\phi}_j} \quad (2)$$

where $\mathbf{S} = (1/m)\mathbf{X}\mathbf{X}^T$ is the data scatter matrix [15,16]. Singular value decomposition (SVD) is performed on the input snapshot matrix $\mathbf{X}$ to calculate the mode matrix $\boldsymbol{\Phi}$. Traditional SVD methods have a computational scaling of $\mathcal{O}(nm^2)$ and require $\mathcal{O}(nm)$ memory [39]. This would be computationally prohibitive considering the amount of training data generated by the free-flight simulations. As in [15], this work leverages instead a streaming SVD algorithm developed by Maulik and Mengaldo [39]. This approximate method reduces the computational cost and the memory needs to $\mathcal{O}(nmh)$ and $\mathcal{O}(nh)$, respectively, where h is a user-defined value to truncate the matrix calculations.[†] As part of the process, the

snapshot matrix is split into smaller batch matrices $\mathbf{A}_i \in \mathbb{R}^{n \times b}$, where b is the number of snapshots associated with each batch. A QR decomposition is applied to the first subset, or batch, of data $\mathbf{A}_0$, resulting in [15]

$$\mathbf{A}_0 = \mathbf{Q}_0 \mathbf{R}_0 \quad (3)$$

A conventional SVD is applied to the generated upper triangular matrix $\mathbf{R}_0$:

$$\mathbf{R}_0 = \mathbf{U}' \mathbf{D}_0 \mathbf{V}_0^T \quad (4)$$

The initial mode matrix $\mathbf{U}_0$ is then given by

$$\mathbf{U}_0 = \mathbf{Q}_0 \mathbf{U}' \quad (5)$$

A value h such that $d < h \ll m$, where d is the number of retained POD modes, is predefined to set the number of singular vectors to preserve with each batch load [15]. The goal is to maintain a significant amount of variance from each batch before optimizing the final number of retained POD modes. For each additional batch, a QR decomposition is applied after the columnwise concatenation of the new batch $\mathbf{A}_i$ to the combined preserved modes and singular values, resulting in

$$[ff^* \mathbf{U}_{i-1} \mathbf{D}_{i-1} \|\mathbf{A}_i] = \mathbf{Q}_i \mathbf{R}_i \quad (6)$$

where a forget factor $ff \in [0,1]$ is specified to vary the effect of previously loaded batches of data on the final result of the algorithm. This application uses an $ff = 1.0$ so that the batches are included equally [39].

The process continues by computing an SVD on the combined data:

$$\mathbf{R}_i = \tilde{\mathbf{U}}_{i-1} \tilde{\mathbf{D}}_{i-1} \tilde{\mathbf{V}}^T \quad (7)$$

The singular values of $\tilde{\mathbf{D}}_{i-1}$ are checked to ensure that they are in descending order. Then $\tilde{\mathbf{U}}_{i-1}$ and $\tilde{\mathbf{D}}_{i-1}$ are sliced to incorporate only the defined h values as specified by the user. The truncated matrix of modes becomes $\hat{\mathbf{U}}_{i-1}$, and the truncated matrix of singular values becomes $\mathbf{D}_i$. The truncated mode matrix is recombined with the orthogonal basis of the incorporated data $\mathbf{Q}_i$, resulting in

$$\mathbf{U}_i = \mathbf{Q}_i \hat{\mathbf{U}}_{i-1} \quad (8)$$

This process is repeated until all batches of data are included. The end result is a matrix $\tilde{\mathbf{U}} \in \mathbb{R}^{n \times h}$ of singular vectors, and a diagonal matrix $\boldsymbol{\Sigma} = \mathbf{D} = \text{diag}(\sigma_1, \dots, \sigma_k) \in \mathbb{R}^{h \times h}$ containing the corresponding singular values σ in descending order. The POD modes $\boldsymbol{\phi}_j$ are found using the d -largest singular vectors of $\tilde{\mathbf{U}}$ who also correspond to the first d columns.

The final number of POD modes d is selected using the relative information content (RIC) defined as

$$\text{RIC}(d) = \frac{\sum_{i=1}^d \sigma_i^2}{\sum_{j=1}^r \sigma_j^2} \quad (9)$$

Desired information content $\delta \in [0,1]$ is selected that preserves d basis vectors that achieve $\text{RIC}(d) \geq \delta$ [15,16,40]. Values of δ can vary depending on the application, although $\delta \geq 0.99$ are commonly found in the literature [15,16,41]. Next, the latent space coordinates are calculated using the retained POD modes by $\mathbf{Z} = \boldsymbol{\Phi}^T \mathbf{X}$ [15,16]. The matrix $\mathbf{Z} = [\mathbf{z}^1, \dots, \mathbf{z}^m] \in \mathbb{R}^{d \times m}$ contains a set of coordinates for each snapshot [16].

C. Interpolation or Regression

The next step in the ROM methodology is to create a parametric mapping of the latent space coordinates $\mathbf{Z}$ to predict unsampled input parameters [15]. Namely, some interpolation or regression

[†]In Maulik and Mengaldo [39] and Dean et al. [15], the user-defined value to truncate matrix calculations is defined by k .

function $g_i: \mathbf{p} \mapsto z_i$ is trained to map some input vector $\mathbf{p} \in \mathbb{R}^m$, where m is the dimension of the input parameter space, to the i th coordinate z_i . With this research, the input $\mathbf{p}$ is composed of vehicle state parameters, such as α , side slip angle β , and M . Note that a total of d independent mappings g_i are trained to predict all the coordinates of the POD latent space, which will be leveraged during the backmapping phase described in Sec. II.D.

Regression techniques used in ROMs include cubic spline [17], radial basis function (RBF) [20,41], kriging [15,16], and artificial neural network (ANN) [42], to enumerate a few. In a previous effort, Dean et al. [15] leveraged a kriging regressor to predict unseen latent coordinates. A kriging model was chosen in particular for its robust training process, ability to handle noise in the training data, and capability to estimate prediction uncertainty. The latter aspect regarding prediction uncertainty is not directly relevant to the current work but is a capability that was identified for future ROM-based trajectory simulation efforts. Dean et al. [15] noted that training kriging models from unsteady CFD results was challenging due to the large amount of training data to process. The training complexity of a kriging model scales with $\mathcal{O}(m^3)$, where m is the number of training samples, i.e., the number of snapshots in the current work. This represents a substantial computational burden considering the current simulation framework generates tens of thousands of samples and the expectation that only more samples will be used in the future. As such, this work implements an approximate or sparse kriging method to alleviate the computational burden of the regression stage of the ROM. The following section briefly summarizes both exact and sparse kriging modeling.

1. Exact Kriging Modeling

A conventional kriging model is a form of Gaussian process regression [43] that represents some quantity of interest as a collection of random variables following a Gaussian distribution. A predictor is formed by conditioning this Gaussian distribution to some observations, i.e., the training data, and taking the mean of the resulting posterior distribution. Using the current nomenclature, the functional representation of the kriging predictor can be generally written as

$$z(\mathbf{p}^*) = \sum_{j=1}^m w_j k(\mathbf{p}^*, \mathbf{p}_j) + \mu(\mathbf{p}^*) + \epsilon \quad (10)$$

where z is the quantity being predicted, $\mathbf{p}^*$ is the input parameter for the prediction, and $\mathbf{p}_j$ are the observed data. The form of Eq. (10) shows that the model is a sum of function $k(\mathbf{p}_i, \mathbf{p}_j)$ multiplied by a corresponding weight w_j . This function is referred to as a covariance or kernel function and captures the dependence between two coordinates in the input space. The result of the predictor is offset by a variable μ that represents the assumed mean of the prior Gaussian distribution. In most situations, μ is either assumed to be constant or to have a simple functional form, such as a multivariate polynomial. For this particular work, μ is represented by a simple linear function. Lastly, ϵ represents the observed noise, which is modeled as $\epsilon \sim \mathcal{N}(0, \sigma_n^2)$, where σ_n^2 is the noise variance.

The covariance function $k(\mathbf{p}_i, \mathbf{p}_j)$ can take several forms, with some being more suited for specific types of problems. Common choices include the Matérn family of functions, which includes the well-known squared-exponential function [43]. This work leverages instead the Matérn 3/2 kernel due to its robustness, which is given by

$$k(\mathbf{p}_i, \mathbf{p}_j) = \sigma_v^2 (1 + \sqrt{3}\Delta) \exp(-\sqrt{3}\Delta) \quad (11)$$

where Δ is a normalized distance defined as

$$\Delta = \sqrt{\sum_{k=1}^b \left(\frac{p_{i,k} - p_{j,k}}{\theta_k} \right)^2} \quad (12)$$

In the above, the variables σ_v^2 and θ_k are the variance and the length scales of the covariance function. Together with the noise variance σ_n^2 , these are often referred to as *hyperparameters*, and their appropriate values must be determined through training. This is commonly done by maximizing the marginal log-likelihood of the Gaussian process. This objective is given by

$$\mathcal{L} = -\frac{1}{2} \tilde{\mathbf{z}}^T (\mathbf{K} + \sigma_n^2 \mathbf{I})^{-1} \tilde{\mathbf{z}} - \frac{1}{2} \log |\mathbf{K} + \sigma_n^2 \mathbf{I}| - \frac{m}{2} \log(2\pi) \quad (13)$$

where $\tilde{\mathbf{z}} = \mathbf{z} - \mu \mathbf{1}$, and $\mathbf{K} \in \mathbb{R}^{m \times m}$ is a covariance matrix whose entries $K_{ij} = k(\mathbf{p}_i, \mathbf{p}_j)$ are the covariance function evaluated for each observation pair. Most of the training computation arises from calculating the inverse and the log-determinant of the matrix $\mathbf{K} + \sigma_n^2 \mathbf{I}$ in Eq. (13). This is typically done using a Cholesky decomposition having a numerical complexity of $\mathcal{O}(m^3)$, which becomes prohibitive for large m . For this reason, kriging is not usually recommended for datasets larger than a few hundred samples [44].

2. Sparse Kriging Modeling

Multiple authors have recognized the scalability issue of kriging and Gaussian process regression and have offered approximate methods to improve their applicability to large datasets (see [45] for a review). Among these solutions are methods using pseudo or inducing points to construct a sparse approximation of a Gaussian process [46–49]. This smaller set of points exists in the same input space as the training data and forms a low-rank approximation of the true covariance matrix. This results in the following Nyström approximation:

$$\mathbf{K}_{mm} \approx \mathbf{Q}_{mm} = \mathbf{K}_{mc} \mathbf{K}_{cc}^{-1} \mathbf{K}_{mc}^T \quad (14)$$

where $\mathbf{K}_{mm} \in \mathbb{R}^{m \times m}$ is the exact covariance matrix as used in Eq. (13), and $\mathbf{Q}_{mm} \in \mathbb{R}^{m \times m}$ is its sparse approximation. This is constructed from the matrix $\mathbf{K}_{cc}$ that holds the covariance between pairs of c inducing points $\mathbf{u}_j \in \mathbb{R}^m$, as well as the matrix $\mathbf{K}_{mc} \in \mathbb{R}^{m \times c}$ representing the covariance between the input samples and the inducing points. Using the Sherman–Morrison–Woodbury formula, one can approximate the inverse of $\mathbf{K}_{mm} + \sigma_n^2 \mathbf{I}$ with

$$(\mathbf{Q}_{mm} + \sigma_n^2 \mathbf{I})^{-1} = \sigma_n^{-2} \mathbf{I} + \sigma_n^{-2} \mathbf{K}_{mc} (\sigma_n^2 \mathbf{K}_{cc} + \mathbf{K}_{mc}^T \mathbf{K}_{mc})^{-1} \mathbf{K}_{mc}^T \quad (15)$$

and using the matrix determinant lemma, its log-determinant with

$$\log |\mathbf{Q}_{mm} + \sigma_n^2 \mathbf{I}| = \log |\mathbf{K}_{cc}^{-1}| + \log |\sigma_n^2 \mathbf{K}_{cc} + \mathbf{K}_{mc}^T \mathbf{K}_{mc}| \quad (16)$$

This sparse approximation results in a complexity of $\mathcal{O}(mc^2)$ that is a significant reduction over the $\mathcal{O}(m^3)$ complexity of the exact solution when $c \ll m$. One should note that the number and location of inducing points $\mathbf{u}_j$ must still be defined. They can be a subset of the observed inputs $\mathbf{p}_j$, be randomly distributed within the input space, or even be determined through some optimization process.

The sparse kriging method implemented in this work is based on the variational free energy (VFE) formulation proposed by Titsias [49]. In this approach, the location of $\mathbf{u}_j$ is optimized jointly with the values of the other hyperparameters $\{\sigma_v^2, \sigma_n^2, \theta_k\}$ of the kriging model during the training. This is made possible by introducing a variational Gaussian distribution that approximates the true posterior distribution. As such, instead of maximizing Eq. (13), the VFE formulation maximizes the variational lower bound of the true marginal log-likelihood, which is given by

$$\begin{aligned} \mathcal{L}_{\text{VFE}} = & -\frac{1}{2} \tilde{\mathbf{z}}^T (\mathbf{Q}_{mm} + \sigma_n^2 \mathbf{I})^{-1} \tilde{\mathbf{z}} - \frac{1}{2} \log |\mathbf{Q}_{mm} + \sigma_n^2 \mathbf{I}| \\ & - \frac{m}{2} \log(2\pi) - \frac{1}{2\sigma_n^2} \text{tr}(\mathbf{K}_{mm} - \mathbf{Q}_{mm}) \end{aligned} \quad (17)$$

The above objective is similar to Eq. (13) except for the last trace term, which acts as a regularization term [49]. Because of this term,

the VFE formulation usually produces more robust models compared to other sparse approximations [50], which tend to overfit. Note that when the inducing points $\mathbf{u}_j$ match exactly the observations $\mathbf{p}_j$, the true log-likelihood of Eq. (13) is recovered. The current sparse kriging model is implemented in Python and leverages the JAX package [51] to automatically differentiate Eq. (17). The training is done using gradient-based optimization with random restarts to prevent the training from getting stuck in a local optimum.

D. Backmapping

Finally, after the regression model is used to predict a point, $\mathbf{p}^*$, a *backmapping* technique is used to project the latent space coordinates back into the high-dimensional space [15]. With POD, backmapping is easily achieved by the linear operation $\mathbf{x}(\mathbf{p}^*) = \Phi \mathbf{z}(\mathbf{p}^*)$, which takes advantage of the direct linear relationship between the latent coordinates and the high-dimensional space [15,16]. This creates a ROM of the form shown in Eq. (18) [15]:

$$\hat{\mathbf{f}}(\mathbf{p}) = \bar{\mathbf{x}} + \sum_{i=1}^d \phi_i z_i(\mathbf{p}^*) = \bar{\mathbf{x}} + \Phi \mathbf{z}(\mathbf{p}^*) \quad (18)$$

E. Evaluating Model Performance

The accuracy of the ROM is evaluated at the ν number of validation points along a selected free-flight validation trajectory. The accuracy of predicting the aerodynamic field using the ROM is determined by calculating the total mean absolute error (MAE) [15]:

$$\text{MAE}_{\text{total}} = \frac{\sum_{j=1}^{\nu} \|\mathbf{x}_j(\mathbf{p}^*) - \mathbf{x}_j^*\|_1}{nv} \quad (19)$$

where $\mathbf{x}_j(\mathbf{p}^*)$ is the predicted field and $\mathbf{x}_j^*$ is the actual field at an unsampled point in the trajectory [15]. Errors in the ROM can be divided into two main sources: compression error and regression error [15]. Some fidelity is lost when the high-dimensional data is projected into the reduced subspace defined by the POD modes. This loss can be measured by compression error and calculated using [15]

$$\text{MAE}_{\text{comp}} = \frac{\sum_{j=1}^{\nu} \|\Phi \Phi^T \mathbf{x}_j^* - \mathbf{x}_j^*\|_1}{nv} \quad (20)$$

where $\Phi \Phi^T \mathbf{x}_j^*$ is the reconstructed field from the original $\mathbf{x}_j^*$ field using the POD modes [15]. If this error is large, it can either indicate that more POD modes are needed to capture the desired fidelity or that the selected DR technique is insufficient to capture the structures present in the high-dimensional data. The other main source of error in a ROM is regression error, which measures the regression model's ability to predict untrained latent space coordinates [15]. It is calculated by

$$\text{MAE}_{\text{reg}} = \frac{\sum_{j=1}^{\nu} \|z_j(\mathbf{p}^*) - z_j^*\|_1}{nv} \quad (21)$$

where $z_j(\mathbf{p}^*)$ is the predicted latent coordinates, and z_j^* is the actual latent coordinates at the unsampled vehicle state [15]. If this error is high, it might be an indication that the regression model of choice is insufficient to predict the trends exhibited by the latent space.

Additionally, axial coefficient C_A , side force coefficient C_Y , normal force coefficient C_N , pitching moment coefficient C_m , yawing moment coefficient C_n , and rolling moment coefficient C_l are calculated by integrating the ROM-predicted pressure and skin friction fields and comparing them to the aerodynamic results generated by FUN3D. Evaluating the ROM-based aerodynamic coefficients is critical to assessing the ROM's ability to capture relevant aerodynamic features in the reduced-order space and predicting these trends at unsampled points. Figure 1 shows the coordinate system for the vehicle, noting the angle of attack α , side slip angle β , bank angle ϕ , and relevant aerodynamic coefficients.

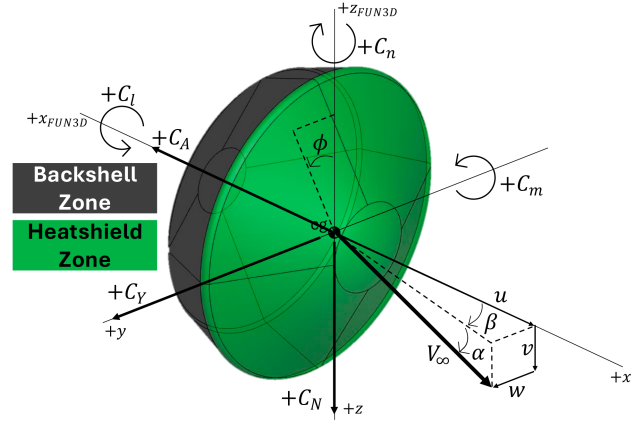


Fig. 1 Vehicle coordinate system.

It should be noted that aerodynamic coefficients are computed along the x - y - z axes shown in the figure. However, FUN3D uses flipped x and z axes internally (denoted x_{FUN3D} and z_{FUN3D}), which are also used for surface field visualization in this study.

III. Application: Genesis Sample Return Capsule

To demonstrate the methodology of the previous section, the Genesis SRC outer mold line was selected to construct an aerodynamic ROM. This capsule geometry has known dynamic instabilities at the low supersonic to transonic Mach regime, leading to an increased tumbling risk during longer flights [52]. Figure 2 shows the dimensions of the ballistic range model utilized to construct the CFD grid, and Fig. 3 shows the near-field and far-field two-dimensional (2D) planar slices of the computational grid used for the CFD-in-the-loop flight simulations. This grid was designed for $-30^\circ \leq \alpha \leq 30^\circ$ and $0.3 \leq M \leq 2.5$, resolving both pressure and skin friction data on the vehicle's surface. These ranges were selected to balance the computational demands of the computational grid and identified trends in the Genesis SRC ballistic range testing shown in [52]. Several refinement regions exist in the grid to capture shear layers, recirculation, wake regions, and the traveling bow shock that occurs as the vehicle decelerates. This results in a total of 55.6 million nodes in the full mesh (240,076 nodes on the vehicle's surface) and about 320.4 million total elements. The CFD simulations were conducted with a detached eddy simulation (DES) turbulence model using a $5 \mu\text{s}$ time step. Additional CFD

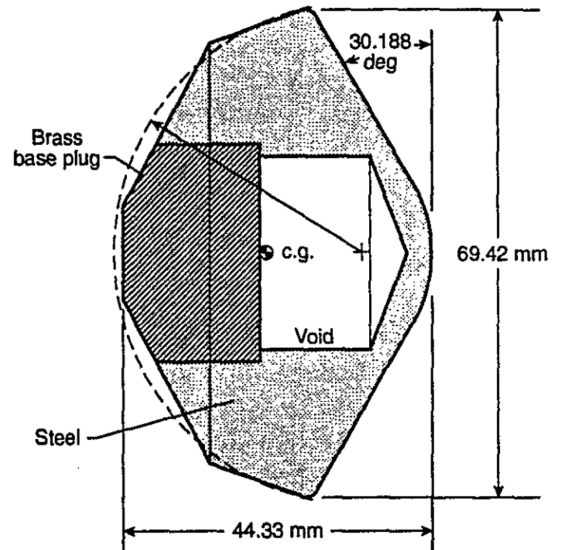


Fig. 2 Genesis SRC ballistic range model (reproduced from [52]).

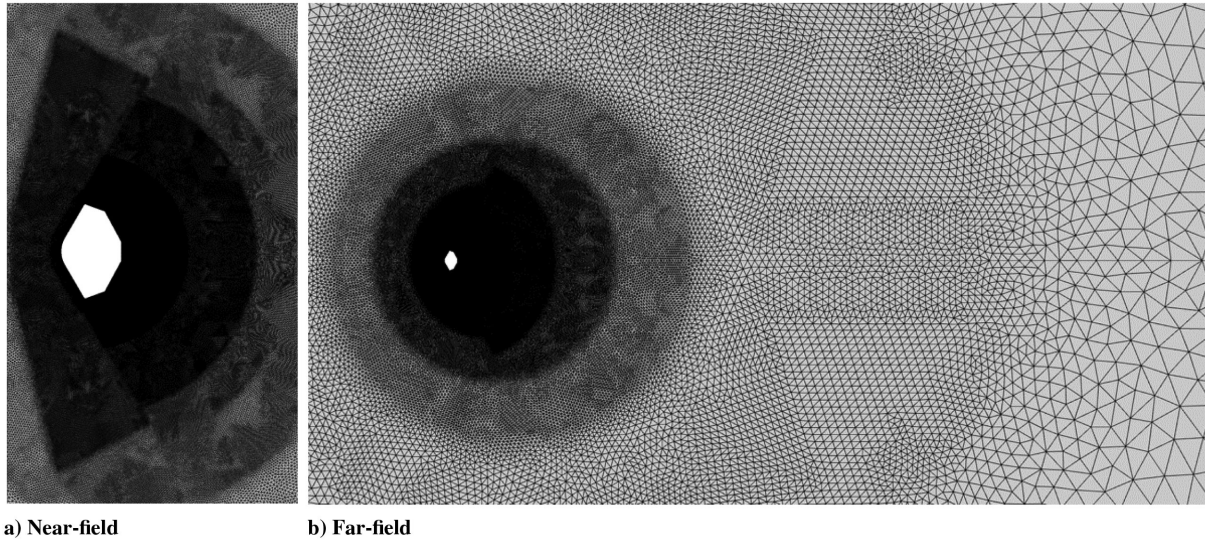


Fig. 3 Two-dimensional slices of free-flight computational grid.

settings include a heuristic van Albada flux limiter (called *hvanal-bada* in FUN3D) and a Dissipated Low-Diffusion Flux-Splitting Scheme (DLDFSS) for flux construction. Comparisons between the CFD-in-the-loop flight simulations and the Cheatwood et al. [52] ballistic range shots showed similar dynamics, indicating the CFD settings and grid were sufficient to capture general trends in the data.

A. Vehicle State Space Sampling

As outlined in Sec. II.A, the free-flight trajectory sampling begins with identifying the relevant state variables. Traditional aerodynamic databases include the atmospheric density ρ , M , α , and β , with the latter sometimes combined into the total angle of attack α_{total} [53–55]. Possible vehicle state parameters to vary in the free-flight simulation include M , pitch angle θ , pitch rate $\dot{\theta}$, yaw angle ψ , yaw rate $\dot{\psi}$, roll angle ϕ , and roll rate $\dot{\phi}$. All these variables can feasibly be varied as initial conditions to the POST2/FUN3D framework. Literature was leveraged to establish the bounds of the sampling space. The initial Mach number was held constant at $M = 1.9$ for each variation in the vehicle state parameters, allowing the vehicle to decelerate down to $M = 0.99$ during the free-flight simulation. This allows future work to compare results with the ballistic range shot at $M_{\text{average}} = 1.4$ from Cheatwood et al. [52].

Initial pitch and yaw angle ranges were selected to be $-15^\circ \leq \theta \leq 15^\circ$ and $-15^\circ \leq \psi \leq 15^\circ$, respectively. Based on the data reported by Cheatwood et al. [52], initial conditions for the reported ballistic range shots tend to be below 10° for all average M numbers. Specifying initial θ and ψ values lower than the bounds of the CFD grid (up to $\alpha \pm 30^\circ$) allows room for oscillations to grow beyond the initial conditions. Including the $\dot{\theta}$ and $\dot{\psi}$ in the sampling is important, as changes in these values can have large implications for the vehicle dynamics. However, $\dot{\theta}$ and $\dot{\psi}$, or even $\dot{\alpha}_{\text{total}}$, are not explicitly reported in the Genesis SRC literature. Additionally, the initialization procedure for the POST2/FUN3D framework does not currently allow the explicit selection of an initial $\dot{\theta}$ and $\dot{\psi}$. To simulate a free-flight trajectory with the POST2/FUN3D framework, a series of initialization steps are conducted: 1) generate a converged FUN3D steady simulation of the static vehicle; 2) generate a converged FUN3D unsteady simulation of the static vehicle; and 3) generate a forced motion simulation to bridge the gap between static simulation conditions and initial conditions of free-flight simulations. The challenge arises in the third step, where a forced motion trajectory is generated to match an input amplitude, frequency, and phase. This process can account for both pitch and yaw forcing functions. Amplitude values are selected based on the desired θ and ψ values. Frequency is determined using the natural frequency of the vehicle at the initial M using Eqs. (22) and (23) together with the

Table 1 Genesis SRC parameter summary

Variable	Value
M_i	1.9
ρ , kg/m ³	1.213
a_{ref}	343.11
$C_{m,\alpha}$	−0.0815
S_{ref} , m ²	0.00378
L_{ref} , m	0.06942
I_{yy} , kg · m ²	1.99E−04
ω_n , rad/s	166.43

vehicle parameters outlined in Table 1. A $C_{m,\alpha}$ value was selected using data reported in Cheatwood et al. [52]. Phase can be visualized as a proxy to varying $\dot{\theta}$ and $\dot{\psi}$ for the vehicle. By varying the phase angle between 0° and 90° and combining that with both negative and positive amplitudes, a range of $\dot{\theta}$ and $\dot{\psi}$ can be achieved.

$$f_n = \frac{\omega_n}{2\pi} \quad (22)$$

$$\omega_n = \sqrt{\frac{-C_{m,\alpha} \hat{q} S_{\text{ref}} L_{\text{ref}}}{I_{yy}}} \quad (23)$$

Roll angles and rates were not reported in the ballistic range testing of the Genesis SRC. Estimated roll rates can be calculated using the spin rate reported for the full-scale Genesis SRC capsule during flight provided by Desai and Cheatwood [56]. This results in a ballistic range model roll rate nearing $2000^\circ/\text{s}$. A roll rate this high aligns with the work of Ernst et al. [33], where roll rates were seen up to $-1860^\circ/\text{s}$ for the SIAD ballistic range shot. These results also show that roll rates remain relatively constant throughout a flight. Therefore, to capture roll effects in the large free-flight sampling space of this vehicle, a significant amount of samples would be required. Therefore, the vehicle will be free-to-roll during the free-flight simulations, but ϕ and $\dot{\phi}$ are not included as DoE sampling variables.

Table 2 summarizes the DoE variables and ranges that were simulated using the POST2/FUN3D framework. A Latin hypercube sampling scheme was selected, resulting in 25 trajectories that were simulated. While the CFD simulation time step was $5 \mu\text{s}$, surface pressure and skin friction shear data were exported every $50 \mu\text{s}$ to

Table 2 Initialization state variable ranges

Variable	Minimum	Maximum
Mach	—	1.9
Frequency, Hz	—	46.49
Pitch amplitude, deg	−15	15
Pitch phase, deg	0	90
Yaw amplitude, deg	−15	15
Yaw phase, deg	0	90

save computational time and storage. Table 3 summarizes the initial conditions of the sampled trajectories, including their ID numbers.

The ROMs are constructed utilizing the data from 24 of the available trajectories. The free-flight trajectory labeled as 113 was omitted from the training of the ROM to be used as the main validation case for ROM capabilities. This process was repeated using trajectory 102 as the left-out case for additional validation. Figure 4 shows the vehicle state space spanned by the capsule throughout the trajectories. The sampled free-flight trajectories cover a wide range of angles and rates the vehicle may encounter in this flight regime. While there are likely still sparsely sampled regions, most notably in the hyperdimensional spaces that result in a star-like pattern (see Pitch

snapshot matrices. It is important to note that when creating a skin friction ROM, the components $\{C_{F_x}, C_{F_y}, C_{F_z}\}$ are combined in the snapshot matrix by stacking the data in the n -dimension.

B. Data Transformations

To enable comparison to literature values and simplify the ROM analysis, the data used in the ROM need to be in a consistent reference frame. The surface data output by FUN3D for each trajectory simulation is in the FUN3D inertial reference frame [57]. Yet entry vehicle aerodynamics are often reported in the body frame, utilizing aerodynamic coefficients like the axial, normal, and pitching moment coefficients. To ease the computational burden, the untransformed and untranslated vehicle grid is used to calculate the aerodynamic coefficients. This is referred to as the origin grid. From this grid, the surface normal vectors are calculated, and no transformation matrix is required to integrate C_p into pressure forces. As for viscous forces, a transformation is required to map the skin friction shear vector $\{C_{F_x}, C_{F_y}, C_{F_z}\}$ into the body frame. It is generally simpler to have the ROM prediction in the body frame, and as such, the C_F data from FUN3D is transformed before the DR step. This is achieved with the following R_{IB} rotation matrix [31]:

$$R_{IB} = \begin{bmatrix} \cos \psi_f \cos \theta_f & \cos \psi_f \sin \phi_f \sin \theta_f - \cos \phi_f \sin \psi_f & \sin \phi_f \sin \psi_f + \cos \phi_f \cos \psi_f \sin \theta_f \\ \cos \theta_f \sin \psi_f & \cos \phi_f \cos \psi_f + \sin \phi_f \sin \psi_f \sin \theta_f & \cos \phi_f \sin \psi_f \sin \theta_f - \cos \psi_f \sin \phi_f \\ -\sin \theta_f & \cos \theta_f \sin \phi_f & \cos \phi_f \cos \theta_f \end{bmatrix} \quad (24)$$

Rate vs Yaw Rate), the sampled space is sufficient to evaluate the free-flight ROM methodology. Each point in this scatterplot matrix represents a sampled pressure and skin friction distribution on the surface of the vehicle that a ROM was trained with. Four ROMs are generated in this research: a heatshield C_p ROM, a heatshield C_F ROM, a backshell C_p ROM, and a backshell C_F ROM. The methodology for constructing each ROM is mostly identical, with small changes noted in the relevant sections. Table 4 summarizes the size of the resulting

where θ_f , ψ_f , and ϕ_f are the pitch, yaw, and roll angles in the body reference frame defined by FUN3D. Additionally, due to the nature of the POST2/FUN3D framework, a correction factor must be applied to the aerodynamic coefficients to account for changing freestream conditions [31]. This is achieved by multiplying the results by Q/Q_{ref} , where Q is the dynamic pressure at the specific time step of interest, and Q_{ref} is the reference dynamic pressure in FUN3D [31].

Table 3 Initial conditions for CFD-in-the-loop flight simulation cases

Job ID	Forcing function inputs				Translated POST2/FUN3D inputs			
	Pitch amplitude	Pitch phase	Yaw amplitude	Yaw phase	Pitch angle	Pitch rate	Yaw angle	Yaw rate
101	2.82	19.59	1.07	26.66	0.94	453.64	0.48	163.49
102	1.48	27.33	3.91	75.99	0.68	224.53	3.80	161.99
103	4.13	56.90	2.04	48.28	3.46	385.78	1.52	231.47
104	−1.26	62.87	−7.26	34.21	−1.12	−98.38	−4.08	−1026.76
105	9.76	6.07	−1.10	29.21	1.03	1659.28	−0.53	−163.47
106	−11.05	3.12	0.12	56.24	−0.60	−1886.20	0.10	11.78
107	−2.95	45.68	−10.33	20.69	−2.11	−352.49	−3.65	−1651.77
108	7.87	32.08	5.43	52.46	4.18	1140.14	4.31	564.27
109	−3.00	83.09	10.48	89.10	−2.98	−61.75	10.48	28.18
110	12.03	33.89	−1.97	41.43	6.71	1707.22	−1.30	−250.37
111	−5.87	17.77	8.22	14.44	−1.79	−955.42	−2.05	−1359.66
112	0.23	24.52	13.93	13.61	0.10	35.95	3.28	2315.56
113	5.99	42.36	8.87	80.77	4.03	756.49	8.75	242.46
114	−7.15	48.18	7.71	39.27	−5.33	−815.55	4.88	1016.22
115	−12.43	11.85	−3.28	74.89	−2.55	−2079.41	−3.17	−146.15
116	4.73	39.26	11.45	3.27	2.99	626.45	0.65	1951.40
117	12.71	76.93	4.44	4.33	12.38	491.41	0.34	739.38
118	−5.25	60.34	−14.19	62.24	−4.57	444.50	−12.56	−1126.74
119	−8.50	67.61	−11.45	83.31	−7.86	−553.71	−11.38	−225.99
120	−14.89	80.44	−5.35	66.86	14.68	−422.55	−4.92	−347.83
121	10.92	74.13	−9.50	68.44	10.50	510.49	−8.84	−586.98
122	7.21	53.56	13.30	22.93	5.80	732.70	5.18	2083.08
123	14.62	70.23	−6.35	45.56	13.75	845.21	−4.53	738.20
124	−12.66	10.79	9.09	9.60	−2.37	−2126.18	1.51	1530.39
125	−9.89	88.98	−12.73	60.39	−9.89	−30.01	−11.07	−1059.37

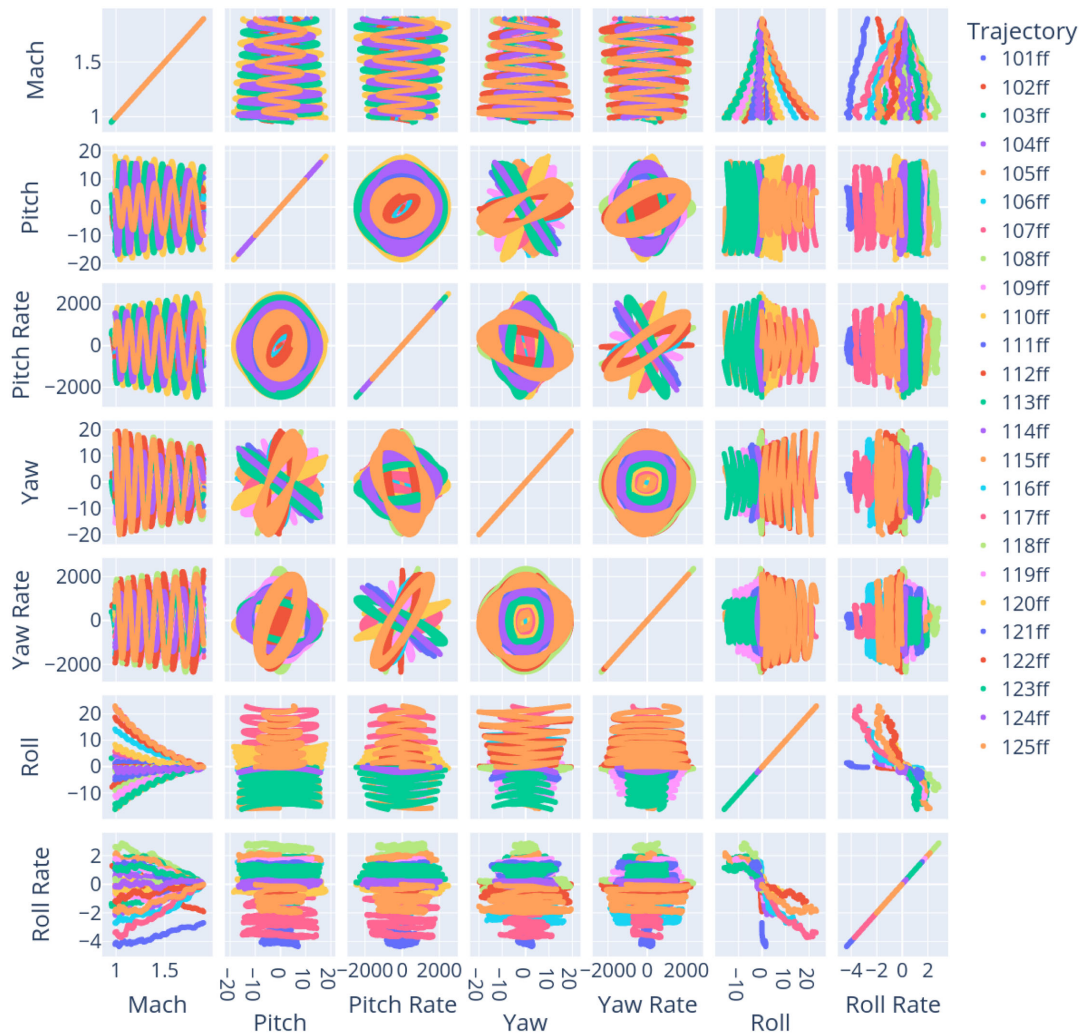


Fig. 4 Vehicle state space for sampled free-flight CFD-in-the-loop flight simulations.

Table 4 ROM snapshot matrix dimensions

Data type	Degrees of freedom, n	Snapshots, m
<i>Heatshield</i>		
C_P	85,452	136,304
C_F	256,356	136,304
<i>Backshell</i>		
C_P	154,624	136,304
C_F	463,872	136,304

IV. Results

A. Dimensionality Reduction

1. Relative Information Content

Figure 5 displays the RIC plots for the heatshield and backshell ROMs for C_P . Figure 6 displays the RIC plots for the heatshield and backshell ROMs for C_F . These plots indicate the amount of variance stored in each POD mode and the number of modes required to achieve the user-specified RIC value. The caption of each figure indicates the RIC value for each ROM. Similar to Dean et al. [15],

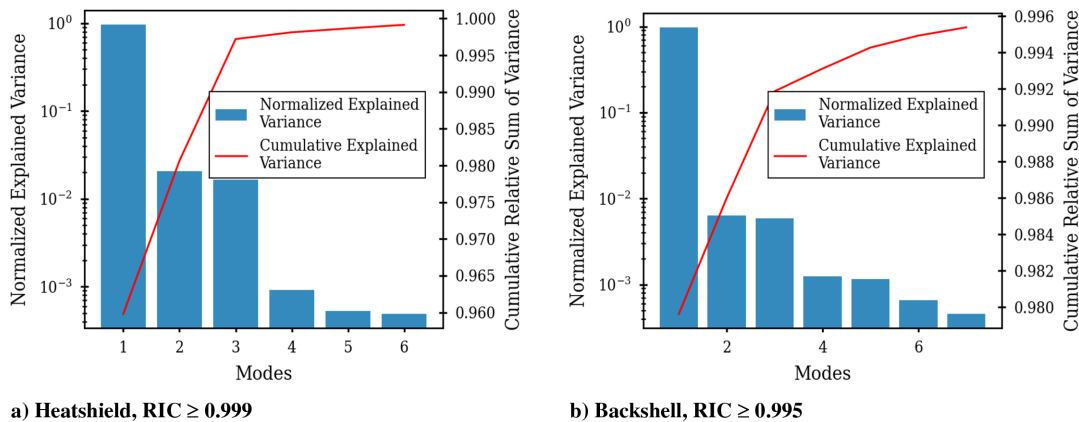
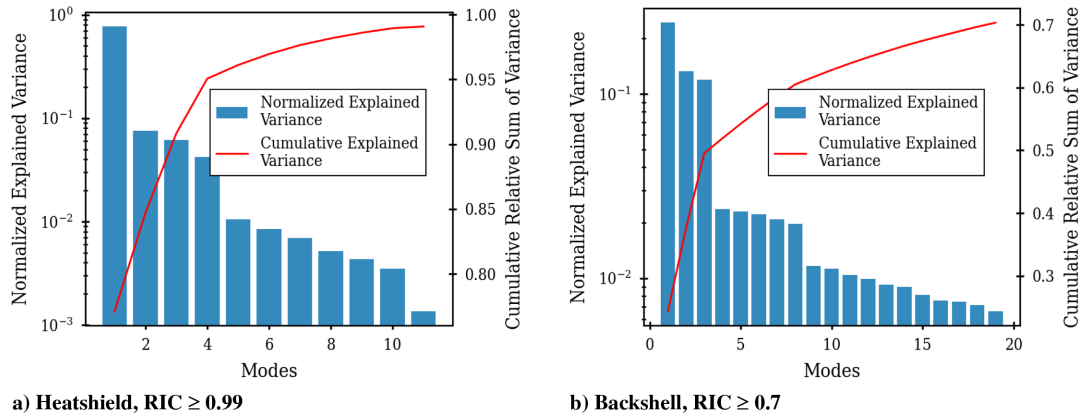


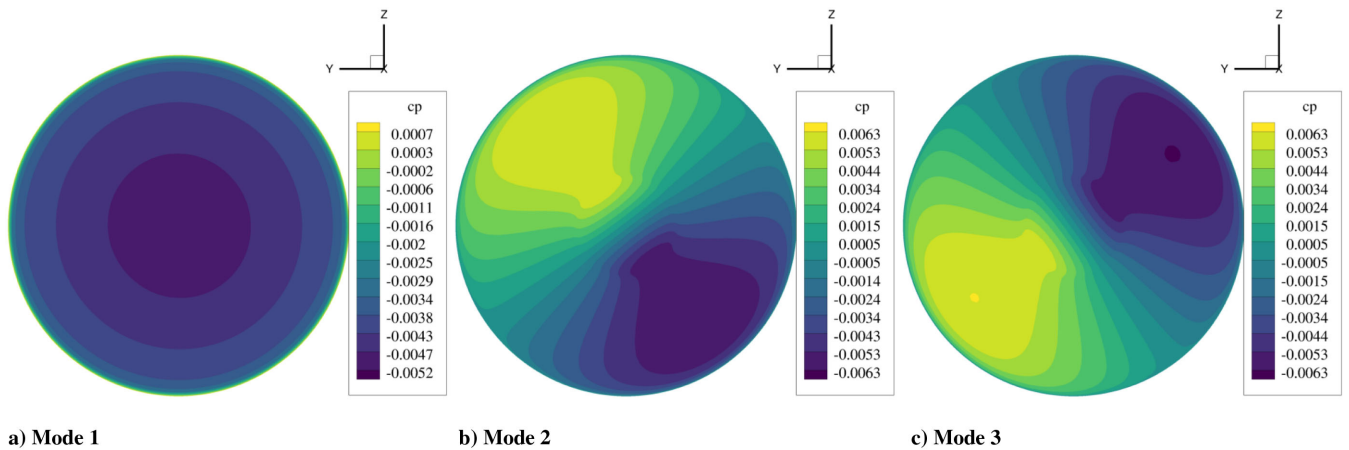
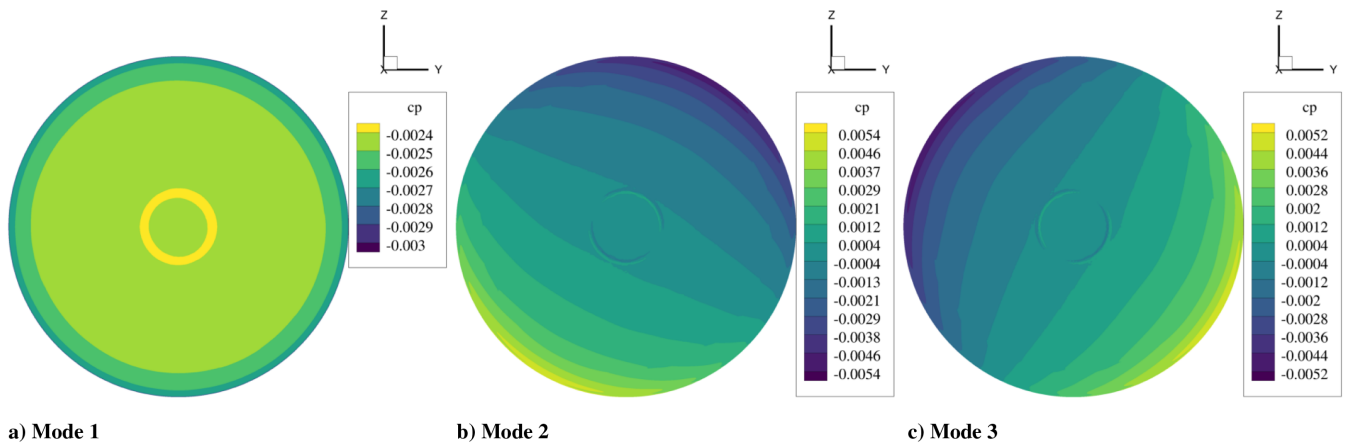
Fig. 5 RIC plots for C_P ROMs.

Fig. 6 RIC plots for C_F ROMs.

POD with streaming SVD significantly compresses the input snapshot matrices. The ROMs average about four orders of magnitude reduction in the degrees of freedom, indicating more distinct, dominant structures occur in the free-flight data. Figure 6b, showing the backshell C_F data, has the lowest RIC value of all the ROMs at 0.7. To achieve an RIC of 0.9, the backshell C_F ROM required over 170 modes. This would be a significant computational burden to regress each mode; therefore, a lower RIC was selected. The high number of modes required to achieve an RIC of 0.9 suggests significant noise or chaos in the data, indicating that a few coherent structures cannot fully explain the variance present.

2. Proper Orthogonal Decomposition Modes

Figure 7 visualizes the three most significant POD modes of the heatshield C_P ROM, and Fig. 8 visualizes the three most significant POD modes of the backshell C_P ROM. POD modes represent the dominant static structures present in the training data. These modes have the same support as the CFD surface grid, allowing the visualization of these modes on the vehicle's surface. The literature notes that POD modes combine both spatial and temporal structures and can be nonphysical [15,58]. Mode 1 on the heatshield and backshell display similar concentric circle patterns, whereas modes 2 and 3 display a striping pattern that appears similar to the pitch or

Fig. 7 Three most significant POD modes for the C_P heatshield ROM.Fig. 8 Three most significant POD modes for the C_P backshell ROM.

yaw directions of the vehicle. Modes 2 and 3 on the heatshield, Figs. 7b and 7c, appear to be similar to the first POD mode of the forced oscillation ROM in Dean et al. [15]. This indicates a similar dominant structure is present in both the forced rotational oscillation data and the free-flight data, likely related to the pitching of the vehicle. The structures of modes 2 and 3 suggest that the side-to-side variations of the field are not aligned with the reference axes of the vehicle. This could imply some coupling in the pitching and yawing dynamics of the vehicle, although Fig. 5a shows that mode 2 is only marginally more important than mode 3.

Figure 9 shows the three most significant POD modes of the C_F heatshield ROM. Recall that to construct the C_F ROM, all vector directions of C_F are combined into a single snapshot matrix. The magnitude of C_F is indicated by the color scale, while white arrows drawn on the surface represent the orientation of the vector field captured by the mode. Mode 1 of the heatshield, Fig. 9a, shows a similar concentric circle phenomenon as the C_P distributions, with the C_F field oriented radially. In this case, the POD mode is a negative of the actual physics, i.e., where the POD vector field is directed inward yet the skin friction vectors should be outward. This makes no difference to the ROM, as the negative of a POD mode is still a valid mode. Modes 2 and 3 show patterns similar to the directions of modes 2 and 3 for the heatshield C_P . The C_F modes tend to be isolated around the shoulder region of the capsule, which can be seen upon close inspection of Fig. 9. Similar trends exist in the three most significant POD modes of the C_F backshell ROM, shown in Fig. 10. Mode 1 shows the C_F vector lines pointing in a similar direction as mode 1 on the heatshield. Although the lines are distinctly more curved, following the shape of the capsule. Modes 2 and 3 on the backshell are also oriented similarly to the C_P backshell

modes, with C_F vectors showing the dominant structures moving around the outer edge of the backshell (see Fig. 2). This phenomenon could align with the direction of the attached flow moving over the three-dimensional (3D) capsule's surface as the vehicle oscillates in that axis. Experimental wind tunnel testing would be necessary to verify the structures identified in this analysis represent physical structures occurring on the vehicle.

3. Latent Space Visualization

The ROM's latent space is sampled by projecting the snapshot matrix into the reduced space spanned by the POD modes. Figure 11 plots the latent coordinate values versus nondimensional time for the three most significant latent dimensions of the C_P heatshield ROM, with colors representing the 24 training trajectories. It is important to remember that the vehicle conducted 6DOF motion to create the training data. Therefore, each point in Fig. 11 can be explained by a set of vehicle state variables. The latent space was plotted against time to represent this hyperdimensional space concisely. From these plots, connections can be drawn between latent space coordinates and vehicle state variables. For example, latent coordinate 1 shows a similar trend as the Mach versus time plot of the vehicle. Latent coordinates 2 and 3 appear to show similar functional relationships to either pitch θ or yaw ψ versus time plots. Using Eq. (20), the MAE of the DR component of the ROM is calculated to be ± 0.0043 , with the mean of the C_P heatshield field being 0.310; this is summarized in Table 5.

Figure 12 plots the latent coordinate values versus the nondimensional time for the three most significant latent dimensions of the C_P backshell ROM. Similar to the heatshield, there are some underlying trends in the latent spaces when plotted through time. While latent

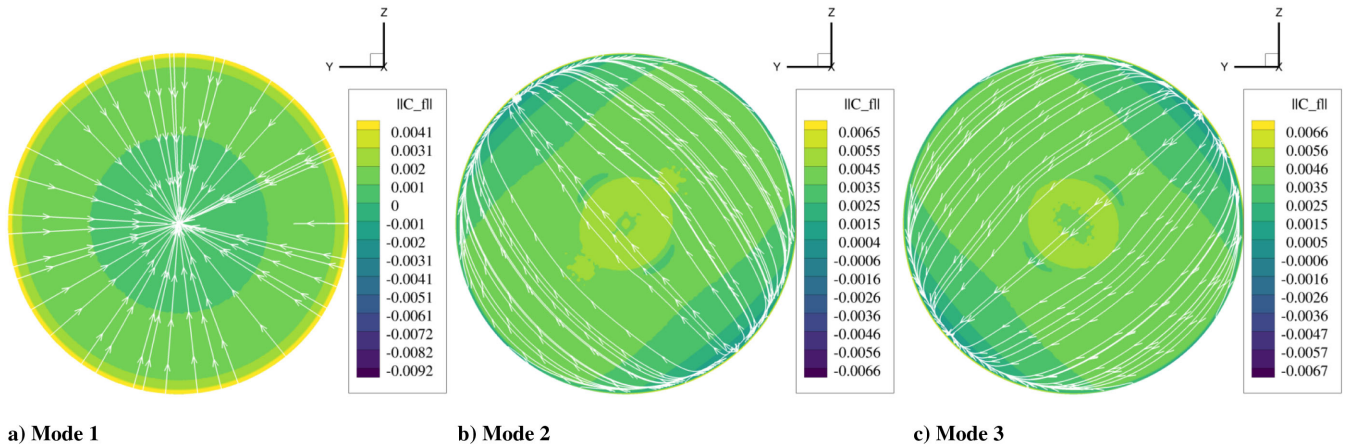


Fig. 9 Three most significant POD modes for the C_F heatshield ROM.

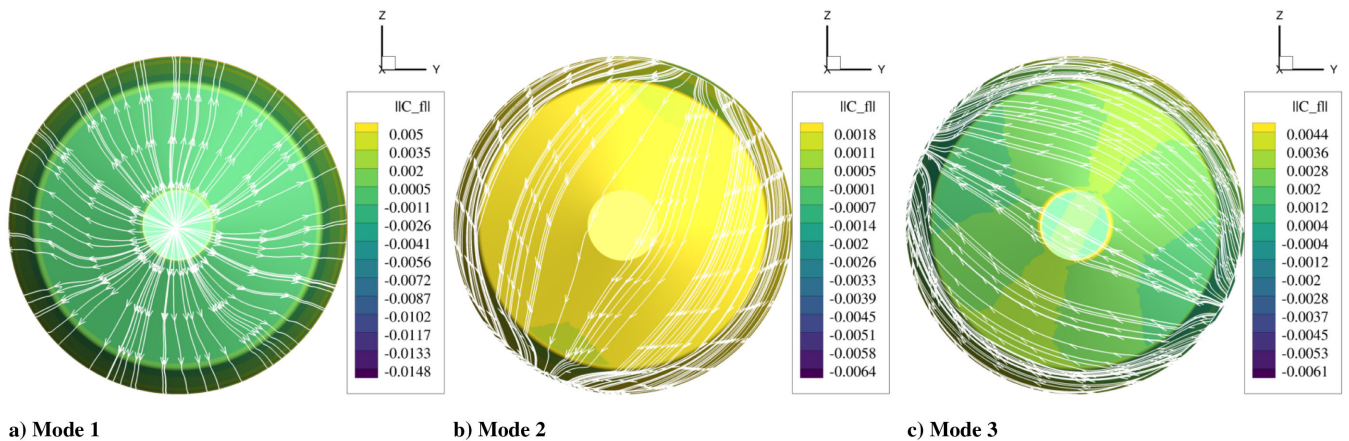


Fig. 10 Three most significant POD modes for the C_F backshell ROM.

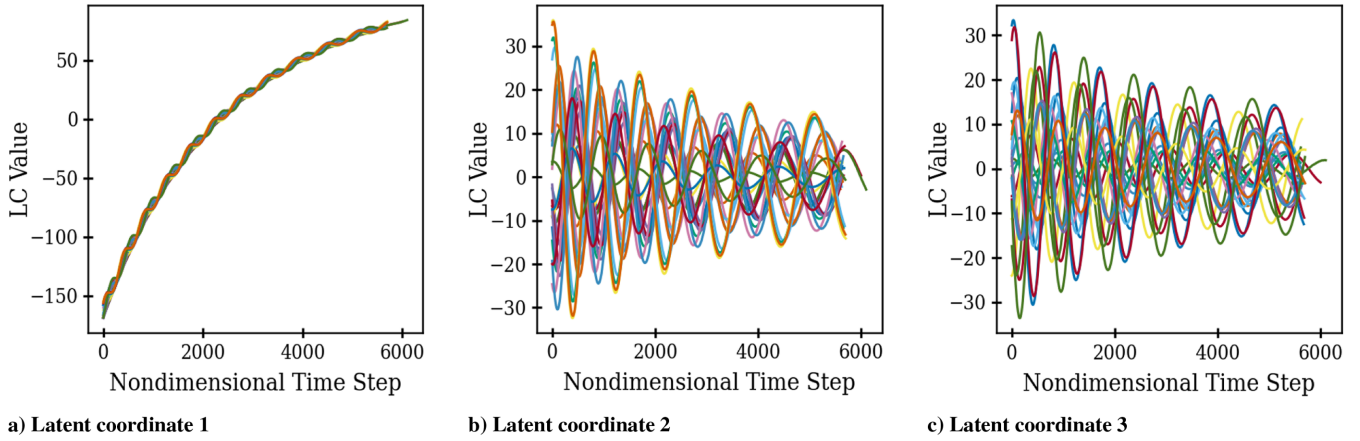


Fig. 11 Three most significant latent spaces for the C_p heatshield ROM.

Table 5 MAE summary

Data type	Compression error	Interpolation error	Total error
<i>Heatshield</i>			
C_p	0.310 ± 0.0043	± 0.548	0.314 ± 0.0036
C_F	$3.37e-4 \pm 2.411e-5$	± 0.0038	$3.41e-4 \pm 2.601e-5$
<i>Backshell</i>			
C_p	-0.184 ± 0.0013	± 0.488	-0.185 ± 0.0047
C_F	$-1.239e-5 \pm 1.595e-5$	± 0.0023	$-1.277e-5 \pm 1.987e-5$

coordinate 1 is mostly linked to the Mach number, there are also oscillations present. Further analyses of these “oscillations” identify the pattern to be a pulse-like change in the C_p field throughout the trajectory, aligning closely with changes in the cross-sectional area of the vehicle along the x -direction. Latent coordinate 2 on the backshell shows oscillations, but the trend also has a distinct “square wave” trend. This trend is emphasized in Fig. 13, where trajectory 115 is plotted with α_{total} . Additionally, Fig. 13 highlights a lagged pressure phenomenon being captured in the latent space. This is most notable in latent coordinates 1 and 2, where the latent space coordinate peaks are shifted slightly from the α_{total} motion of the vehicle, indicated by the dashed black line. Literature notes that this lagged pressure phenomenon exists in experimental wind tunnel tests for blunt-body entry capsules [59]. It is important to note that the distinct structures of the backshell were unlike the structures

seen in Dean et al. [15], which displayed much more chaotic phenomena and turbulent structures. Equation (20) is used to calculate the MAE of the DR component of the ROM. The MAE at each node is ± 0.0013 , while the mean of the C_p backshell field is -0.184 . Table 5 summarizes these values.

Figure 14 plots the latent coordinate values versus nondimensional time for the three most significant latent dimensions of the C_F heatshield ROM. The trends in C_F are similar to C_p ; the main difference is the magnitude of the latent coordinate value. The most significant latent coordinate of C_p ranged around $[-150, 60]$, whereas C_F ranged around $[-0.4, 0.3]$. The C_F -based results likely contribute less to the total aerodynamic coefficient compared to C_p -based values but might still play a key role in the capsule dynamics over time. Using Eq. (20), the MAE of the DR component of the ROM is calculated to be $\pm 2.411e-5$, with the mean of the C_F heatshield field being $3.37e-4$; these values are summarized in Table 5.

Similar to the C_p backshell latent space, more noise or chaos is present in the backshell C_F latent space, which is shown in Fig. 15 for the three most significant latent coordinates. Latent coordinate 1 appears to be highly correlated with M variations, with a distinct bend in the data around time step 2000 that is unlike the other explored latent spaces. Latent coordinate 2 also appears to have a “square-wave”-type feature, with the changes in latent coordinate value being sharper than in C_p . It is hypothesized that the square wave captures an attachment and detachment phenomenon that occurs on the backshell of these vehicles. The trend is likely sharper in the C_F backshell field because it would be related to the shear

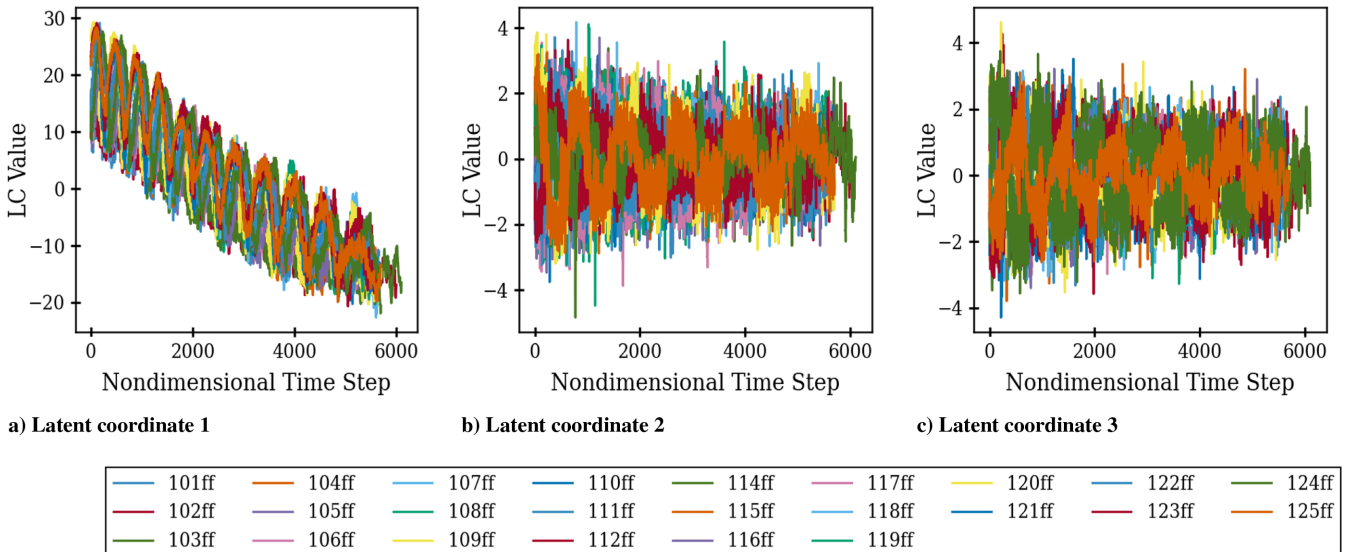


Fig. 12 Three most significant latent spaces for the C_p backshell ROM.

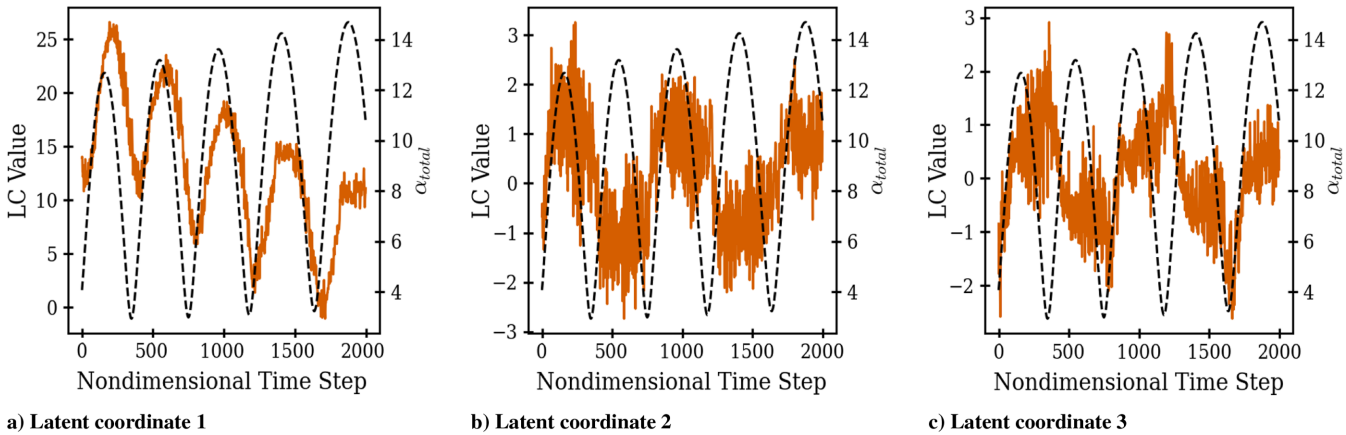


Fig. 13 Zoomed-in version of three most significant latent spaces for the C_P backshell ROM—trajectory 115.

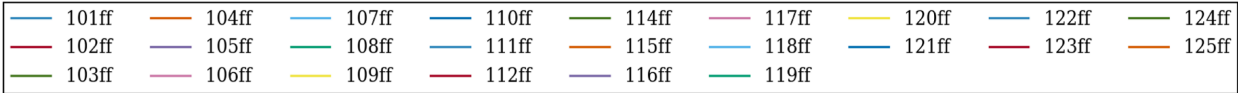
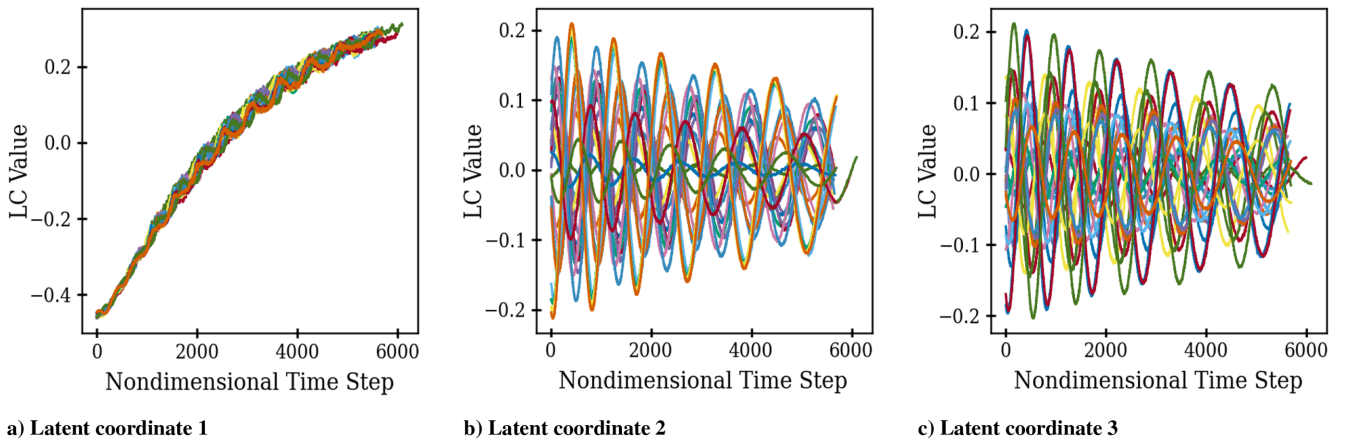


Fig. 14 Three most significant latent spaces for the C_F heatshield ROM.

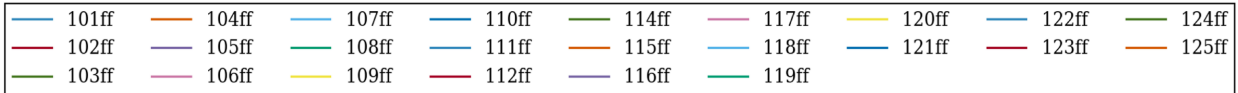
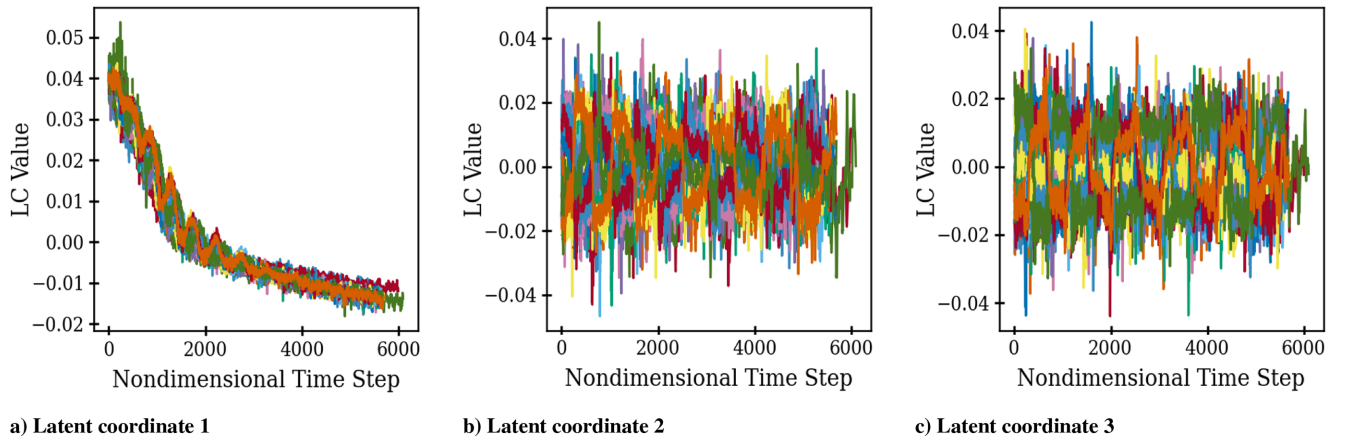


Fig. 15 Three most significant latent spaces for the C_F backshell ROM.

layers in that region. Experimental testing would be required to confidently draw connections between the phenomena observed in the reduced-order space and the actual physics. The MAE is calculated using Eq. (20), resulting in the MAE at each node equal to $\pm 1.595e-5$, with the mean of the C_F backshell field being $-1.239e-5$; these values are summarized in Table 5.

Similar to Dean et al. [15], autocorrelation and frequency domain analysis were applied to the latent space to isolate the frequency content of the data. Autocorrelation, or other time series statistics, can be used to decipher signal correlations in the data [60]. Results for the shown latent spaces consistently showed the dominant frequencies of latent coordinates 2 and 3 to be 19.19 and 18.84 Hz. These values are around the average natural frequency of the entry vehicle between $0.99 \leq M \leq 1.9$. Additionally, power spectral density (PSD) values, specifically in the backshell latent spaces, show high-frequency content that decays at a rate of $f^{-5/3}$, aligning with turbulent theory on energy cascade and the previous work of Dean et al. [15].

4. Reconstructed Aerodynamic Coefficients

By representing the free-flight data with reduced modes, some level of fidelity is inevitably lost. Reconstructing the aerodynamic coefficients from the POD modes and the latent spaces of the training data is a way to evaluate which data structures remain of the high-dimensional training data. The high-dimensional space of the training data is reconstructed using backmapping, where $\mathbf{x}(p^*) = \Phi \mathbf{z}(p^*)$. The resulting low-dimensional form of the input snapshot matrix $\mathbf{X}$ is integrated to produce forces and moments and then nondimensionalized to produce aerodynamic coefficients [15]. Pressure-based aerodynamic coefficient reconstructions were explored, demonstrating similar accuracy as the pressure-based reconstructions from Dean et al. [15]. Figure 16 shows the reconstructed aerodynamic

coefficients for C_F heatshield data for one of the training trajectories. The red lines represent the CFD-based aerodynamic coefficients for C_F heatshield data, and the green lines represent the reconstructed aerodynamic coefficient for C_F heatshield data from the reduced latent space and POD modes. POD sufficiently maps the full-order space into a linear subspace and preserves the majority of the data structures associated with the aerodynamic coefficient responses. Visually, the largest discrepancies appear to be occurring in the rolling moment, C_l (Fig. 16f); however, the magnitude of this coefficient is small. It is possible that the selected number of modes washes out some of the chaotic phenomena associated with the rolling moment. By preserving more modes during dimensionality reduction, more of these phenomena could be captured. However, due to the low magnitude of this coefficient, the benefits of capturing this chaotic phenomenon may not outweigh the computational expense of training additional POD modes.

Figure 17 shows the aerodynamic contributions for C_F backshell of one of the training trajectories. Similar to Fig. 16, these plots indicate that the data structures relevant to the vehicle aerodynamic response are being captured by POD. Additionally, the backshell C_F data only uses 70% of the RIC of the high-dimensional data; therefore, it is significant that the POD modes are capturing these trends with this low RIC value. The rolling moment C_l shows similar trends on the backshell, indicating that the selected modes wash out the low-magnitude rolling moment trend. It is important to recognize that the free-flight trajectories were not initialized by distinct roll angles and roll rates. The roll angles and rates seen here result from the simulations' unsteadiness. It is possible that if trajectories were sampled in those variables, more distinct structures associated with roll would appear in the data, and C_l would be captured in more dominant modes. Aerodynamic coefficients associated with C_p data follow similar trends to the C_F data, as well as mimic the minimal discrepancies shown in Dean et al. [15]. The C_p data also

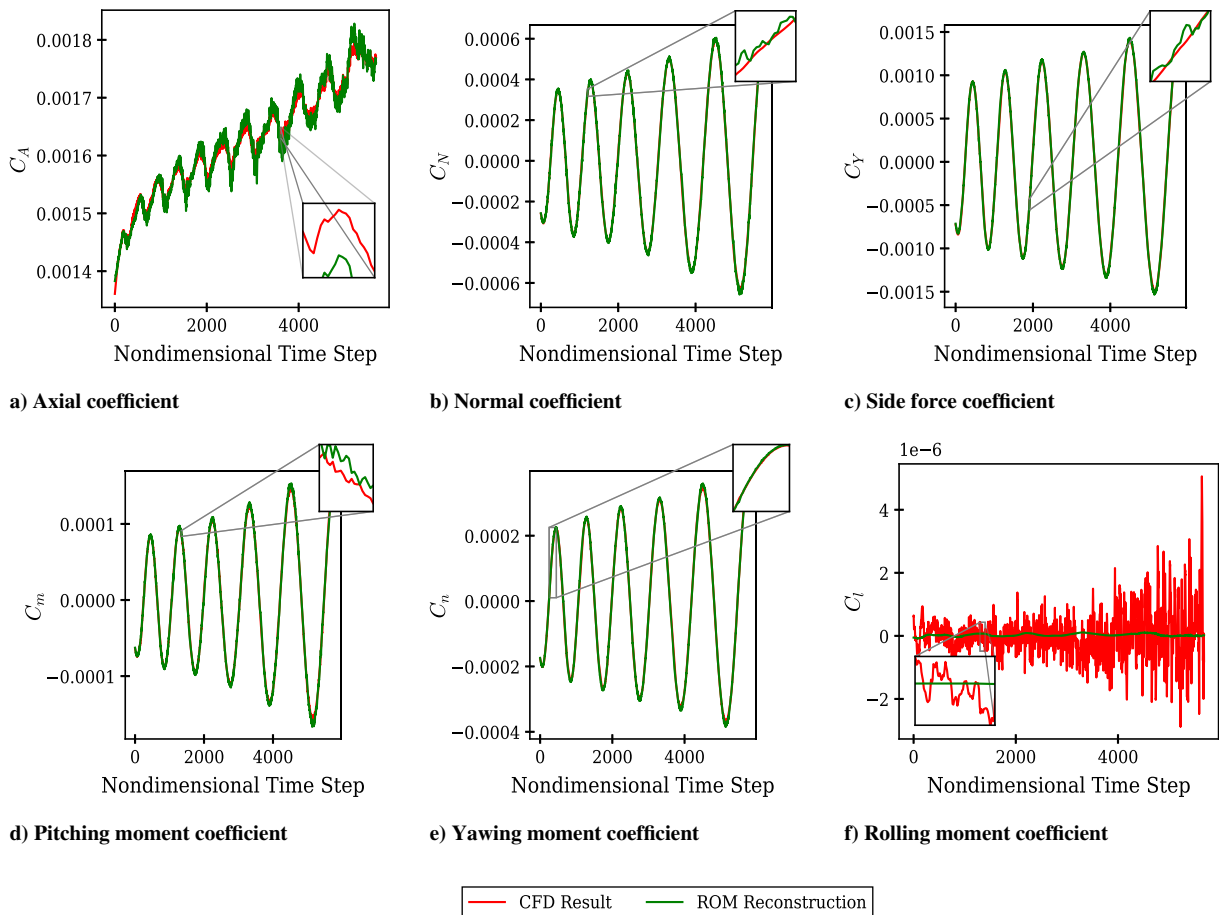


Fig. 16 Actual vs reconstructed heatshield C_F contributions to aerodynamic coefficients—trajectory 118.

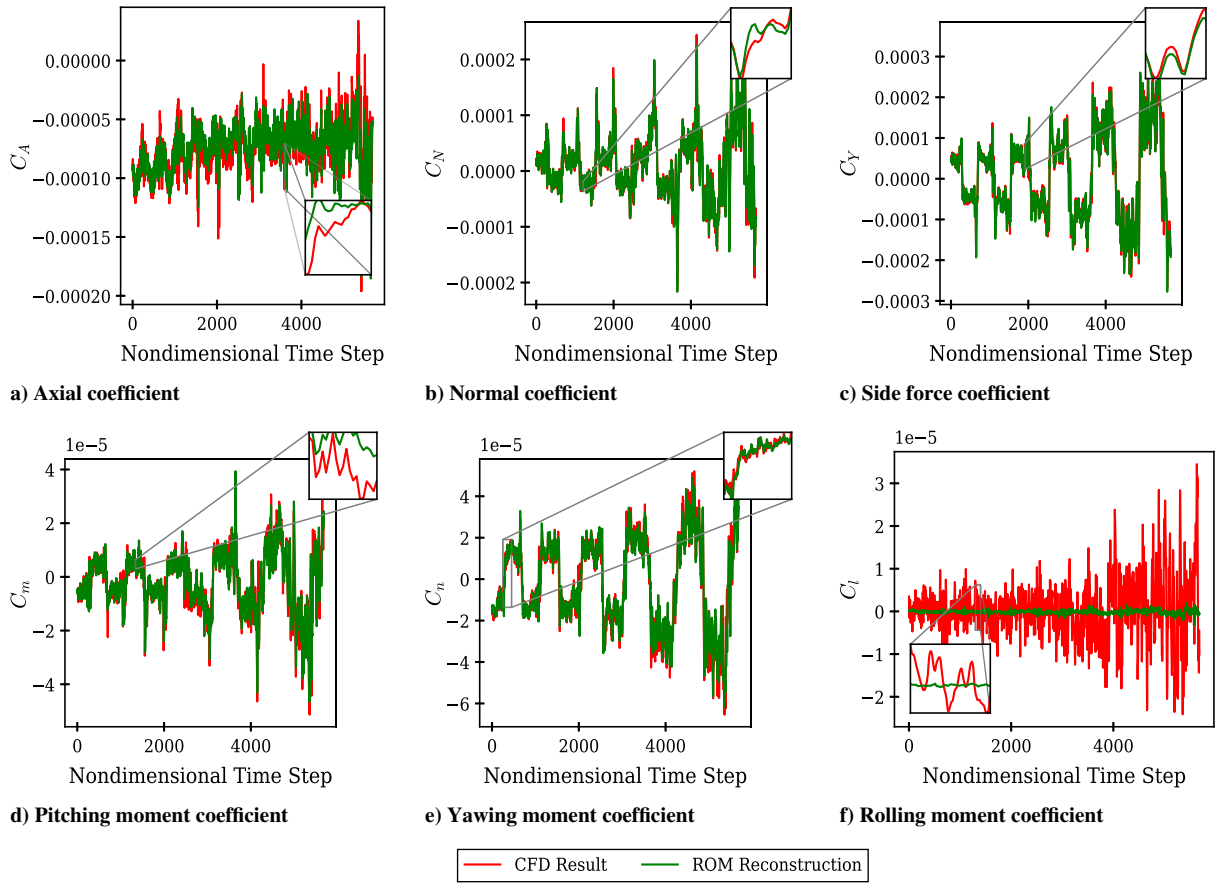


Fig. 17 Actual vs reconstruction of backshell C_F contributions to aerodynamic coefficients—trajectory 118.

shows similar discrepancies in C_l , like those seen in the C_F reconstructions.

B. Predicting an Unsamped Free-Flight Trajectory

1. Predicting the Latent Space

A sparse kriging model using a Matérn 3/2 kernel was trained using the state space variables M , θ , $\dot{\theta}$, ψ , $\dot{\psi}$, ϕ , and $\dot{\phi}$ to map the ROM latent space. The ROMs were tested using the unseen free-flight trajectory resulting from the initial conditions of trajectory 113 (see Table 3). The dynamics of this validation trajectory fall in the middle of the training data set, minimizing extrapolation in predictions. Figure 18 shows the actual versus predicted plots of the three most significant latent spaces for the C_P heatshield ROM.

While predictions were made using $\{M, \theta, \dot{\theta}, \psi, \dot{\psi}, \phi, \dot{\phi}\}$, the data is plotted against time for simplicity. The red points indicate the actual latent space of the validation trajectory, calculated by taking the actual C_P distribution of the heatshield and projecting the data into the latent space via the POD modes. The blue points are the predicted latent space of the ROM. The trained model can accurately predict the latent space of the validation trajectory, with small discrepancies existing in the first latent space. Using Eq. (21), the MAE for this component of the ROM was ± 0.548 . This is a small amount of error relative to the magnitude of latent space values being predicted; Table 5 summarizes this value.

Figure 19 shows the actual versus predicted values of the three most significant latent spaces versus time for the C_P backshell

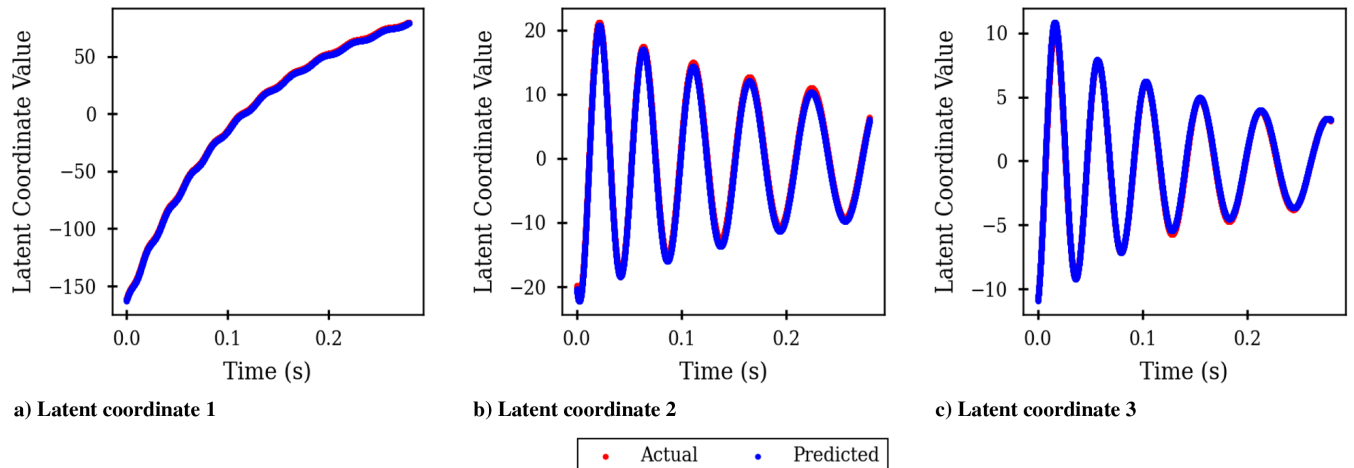


Fig. 18 Actual vs predicted of three most significant latent spaces for heatshield C_P ROM.

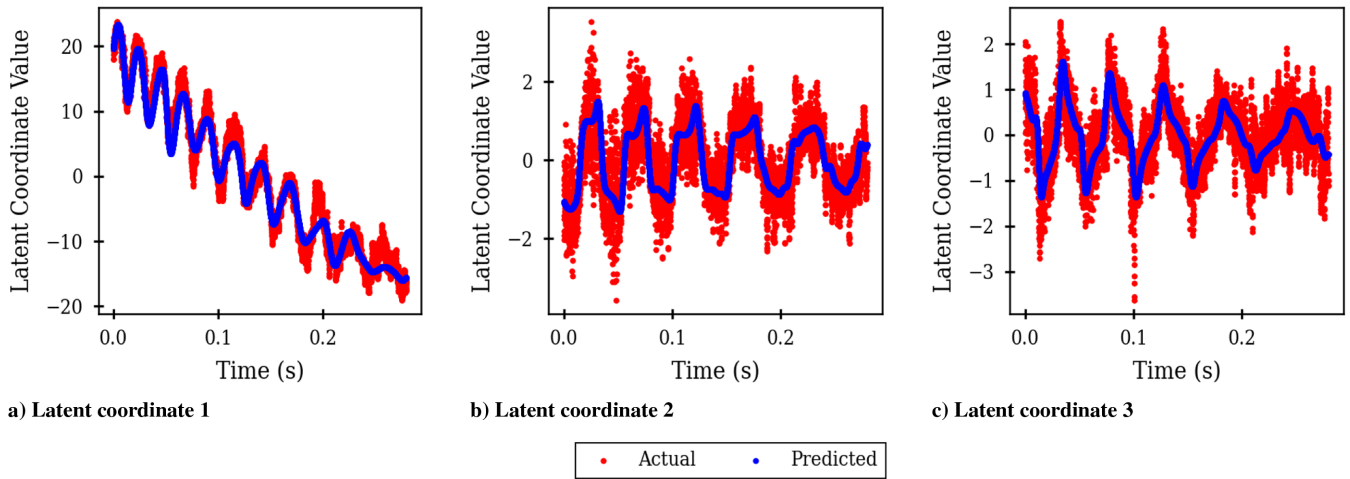


Fig. 19 Actual vs predicted of three most significant latent spaces for backshell C_P ROM.

ROM. The sparse kriging model using the state space variables can capture the mean trends exhibited by the validation trajectory. Additionally, the “square wave” phenomena of latent coordinate 2 (see Fig. 19b) can be visualized more distinctly by the predictions of the regression. The model captures the mean trend of the validation trajectory latent space, with the capability to capture the chaotic phenomena through uncertainty quantification. The MAE of the regression component of the ROM was calculated using Eq. (21). For the C_P backshell latent space, the MAE is ± 0.488 . This is an acceptable margin since the magnitude of the latent space values is large; this is summarized in Table 5.

Figure 20 shows the actual versus predicted values for the three most significant latent spaces for the C_F heatshield. The trends in these latent space predictions are nearly identical to the C_P data, the key distinction is the magnitude of the latent space coordinate. The trained model can accurately predict the validation trajectory in C_F . Larger discrepancies between the actual versus predicted plots can be seen in peaks of the oscillation. It is unclear if these discrepancies are due to poor sampling around the peaks of the validation trajectory or if some significant structures are absent for the selected POD modes. Future analysis would be required to identify the root cause of these errors. The MAE of the regression component of the ROM was calculated using Eq. (21). For the C_F heatshield latent space, the MAE is ± 0.0038 , which is summarized in Table 5.

In contrast, the C_F backshell magnitudes for the three most significant latent spaces shown in Fig. 21 are orders of magnitude smaller than both C_P backshell, and C_F heatshield spaces. Even with the small magnitude, the model predictions are fairly accurate. Similar to C_P backshell predictions, the C_F predictions capture the

mean trend of the data, with the chaotic phenomena likely captured by implementing some uncertainty quantification within the regression. The MAE of the regression component of the ROM was calculated using Eq. (21). For the C_F backshell latent space, the MAE is ± 0.0023 ; this is summarized in Table 5.

2. Predicting Aerodynamic Fields

The C_P and C_F fields can be reconstructed by leveraging linear transformations of the POD modes and the predicted latent coordinates. For each $\{M, \theta, \dot{\theta}, \psi, \dot{\psi}, \phi, \dot{\phi}\}$ combination in the validation trajectory, distribution of C_P and C_F can be calculated for both the heatshield and backshell of the vehicle. An absolute error calculation between the actual and predicted aerodynamic field at each step of the trajectory was calculated. Then, the absolute error of each field at each time step of the trajectory is averaged together to plot the error in the field across the trajectory into a 2D plot. Figure 22 displays the average absolute error of the C_P ROMs. The heatshield C_P fields have randomized errors, indicating the ROM has sufficiently captured the dominant features in the data. The backshell C_P fields appear to have the largest errors around the shoulder region of the vehicle, in a direction that aligns with POD mode 2. The large error is likely a result of the distribution of points present in the latent space. The total MAE of the C_P ROMs was calculated using Eq. (19). For the heatshield, the MAE at each node was calculated to be ± 0.0036 , with the predicted C_P mean equal to 0.314. For the backshell, the MAE at each node was calculated to be ± 0.0047 , with the predicted C_P mean equal to -0.185 ; this is summarized in Table 5.

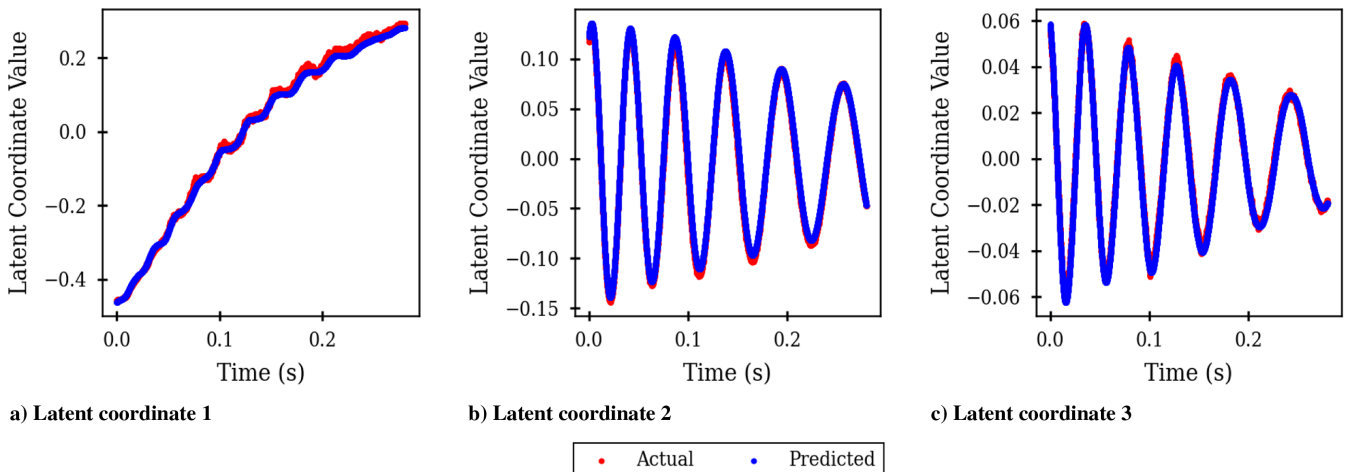


Fig. 20 Actual vs predicted of three most significant latent spaces for heatshield C_F ROM.

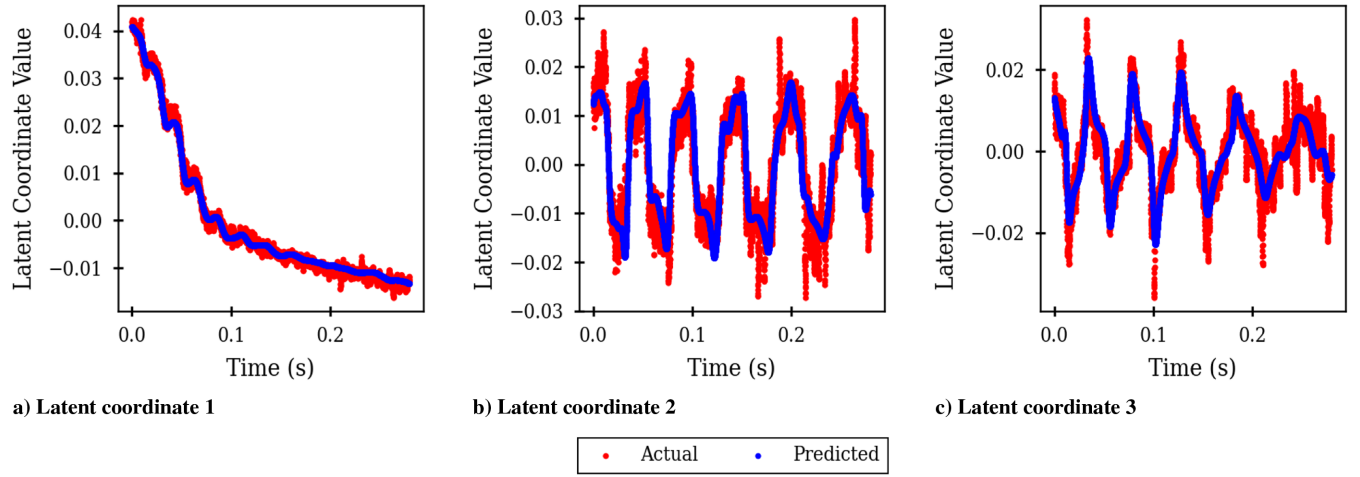


Fig. 21 Actual vs predicted of three most significant latent spaces for backshell C_F ROM.

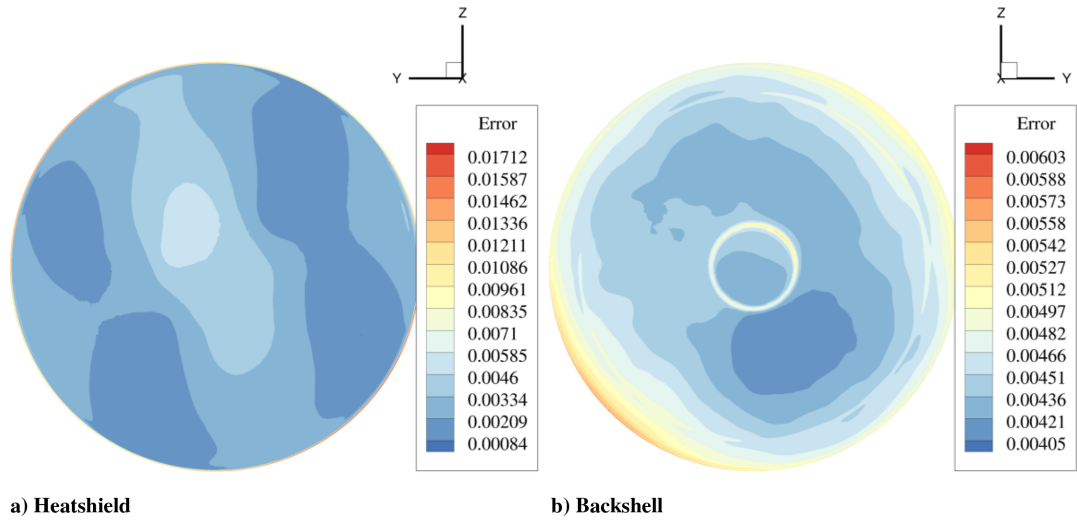


Fig. 22 Average absolute error in C_P across validation trajectory.

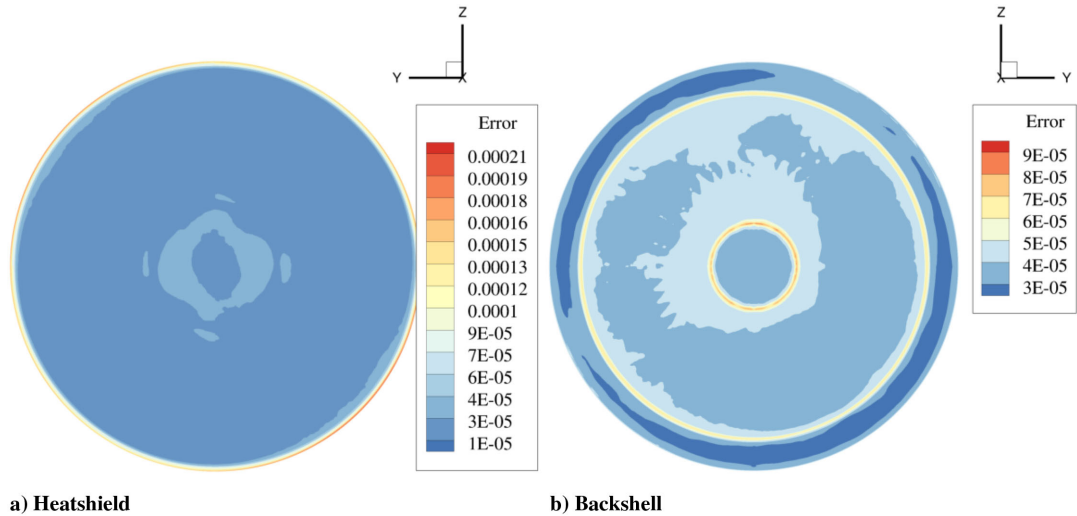


Fig. 23 Average absolute error in C_F across validation trajectory.

Figure 23 shows the average absolute error of the C_F ROMs. On average, the largest errors in these plots tend to be isolated around the sharp edges of the vehicle's outer mold line. For the heatshield, this is around the shoulder region of the vehicle. For the backshell,

this is around the second angled region and top edge of the vehicle. The total MAE of the C_F ROMs was calculated using Eq. (19). For the heatshield, the MAE at each node was calculated to be $\pm 2.601e-5$, with the predicted C_F mean equal to $3.41e-4$. For

the backshell, the MAE at each node was calculated to be $\pm 1.277e-5$, with the predicted C_F mean equal to $1.987e-4$; this is summarized in Table 5.

3. Predicting Aerodynamic Coefficients

Each predicted aerodynamic field can be integrated to generate aerodynamic coefficients to evaluate the ROM's abilities to predict the dynamic aerodynamics of the entry capsule. For each ROM, C_A , C_Y , C_N , C_m , C_n , and C_l were calculated, comparing the ROM-predicted values to the actual values produced from the CFD aerodynamic fields. For conciseness, only C_A , C_N , and C_m are reported. Relevant trends in the other aerodynamic coefficients are noted when applicable. The contributions of C_P , C_F , heatshield, and backshell to the aerodynamic coefficients remained separated to isolate the accuracy of each ROM's predictions within each region and field. Figure 24 shows the aerodynamic coefficients resulting from the heatshield C_P distributions. The red line indicates the CFD-based coefficient for the particular data region of interest. For example in Fig. 24, the red line is the CFD-based coefficients resulting from the C_P heatshield region. The blue line is the resulting aerodynamic coefficients from the ROM-predicted fields. The ROM can accurately predict many of the aerodynamic coefficients, utilizing only the POD modes and latent coordinates predicted with $\{M, \theta, \dot{\theta}, \psi, \dot{\psi}, \phi, \dot{\phi}\}$. Small discrepancies exist at the peaks of C_N and C_m ; however, the main aerodynamic trend is captured accurately with the ROM.

Figure 25 shows the aerodynamic coefficients resulting from the backshell C_P distributions. The backshell C_P contributions are noticeably more chaotic, but the mean trends are still captured by the ROM. The chaotic phenomena of the backshell might be captured in lower dimensions in the latent space; thus, including those spaces may improve the predictive performance. However, it is important to evaluate whether or not increasing the dimensionality of the ROM is required to capture the dynamic stability phenomena of interest. Additionally, future work can explore how the uncertainty bounds of the latent space captured by sparse kriging are projected into the aerodynamic coefficients.

Figure 26 shows the aerodynamic coefficients resulting from the C_F heatshield ROM. Similar to the C_P heatshield results, the C_F heatshield aerodynamic coefficients can be predicted accurately for the validation trajectory, with the largest discrepancies occurring in C_A . Similar to C_P , the ROM was able to capture the oscillating trends in C_N and C_m , with slight discrepancies occurring in the peak values. These plots show that the heatshield C_F ROMs contribute several orders of magnitude smaller aerodynamic coefficients than their C_P counterparts. This raises questions regarding the necessity to model C_F trends with a ROM to capture the dynamic stability phenomena of an entry capsule.

Figure 27 shows the aerodynamic coefficients resulting from the C_F backshell fields. At an RIC of 70%, the backshell C_F ROM captures the mean trends. The largest discrepancies between the CFD-based result and the ROM predictions are in C_A , which was a

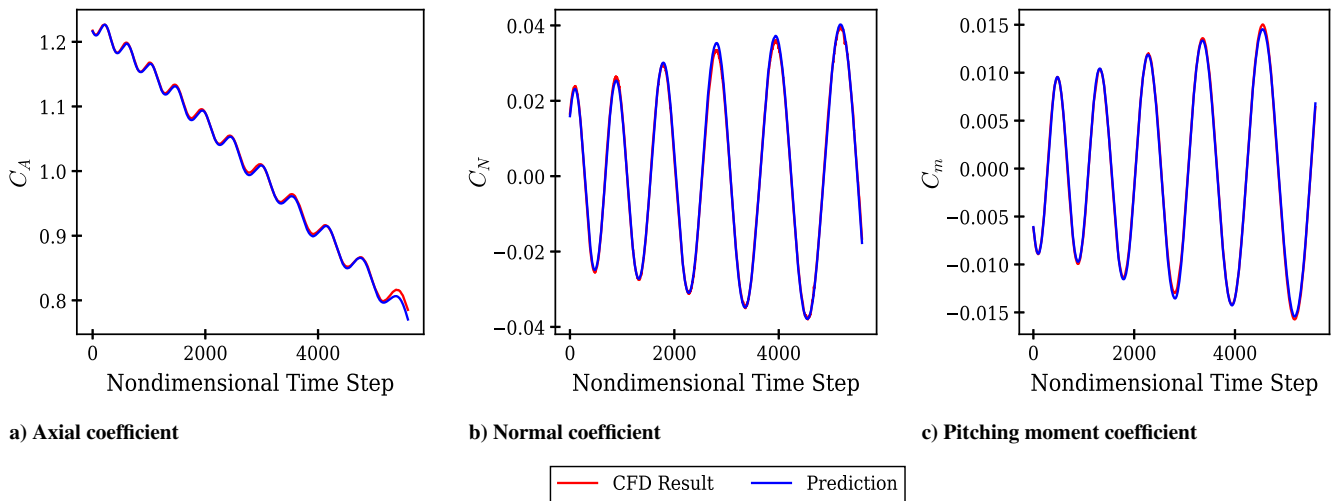


Fig. 24 Actual vs predicted C_P contributions to aerodynamic coefficients across trajectory 113—heatshield.

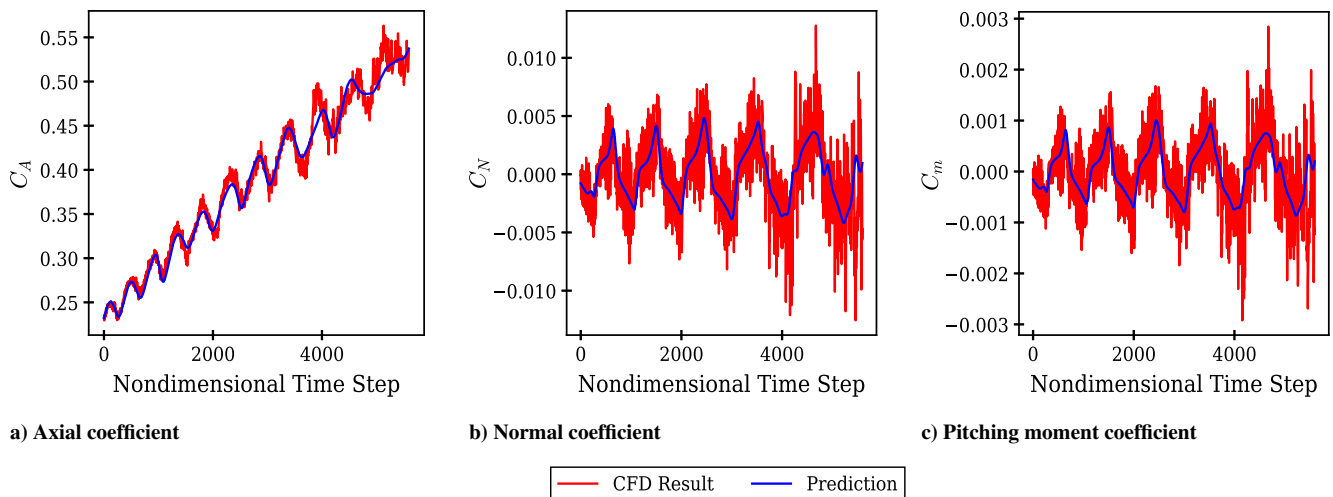


Fig. 25 Actual vs predicted C_P contributions to aerodynamic coefficients across trajectory 113—backshell.

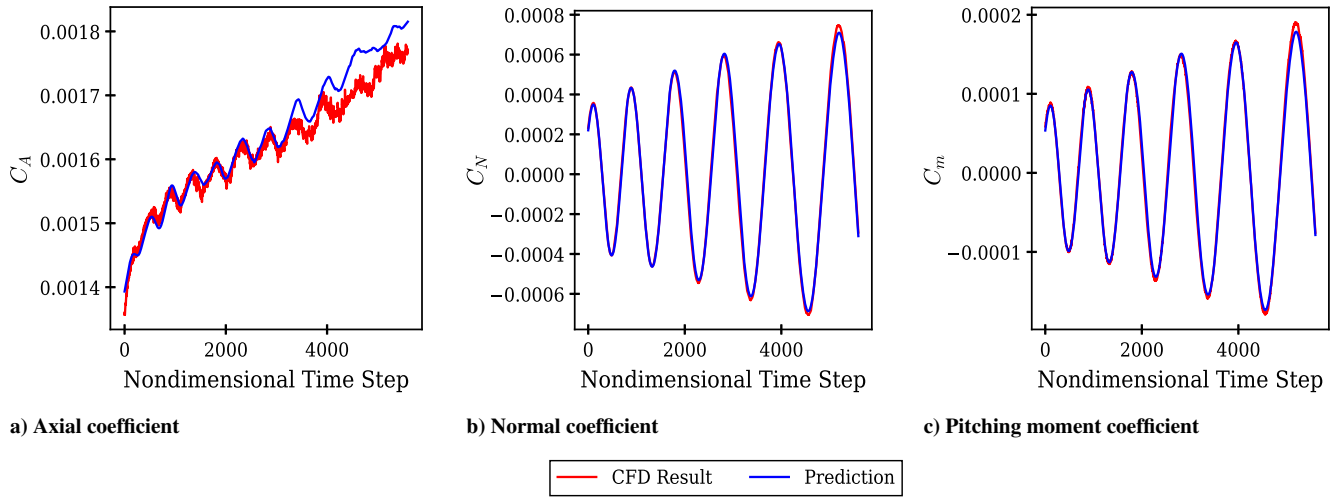


Fig. 26 Actual vs predicted C_F contributions to aerodynamic coefficients across trajectory 113—heatshield.

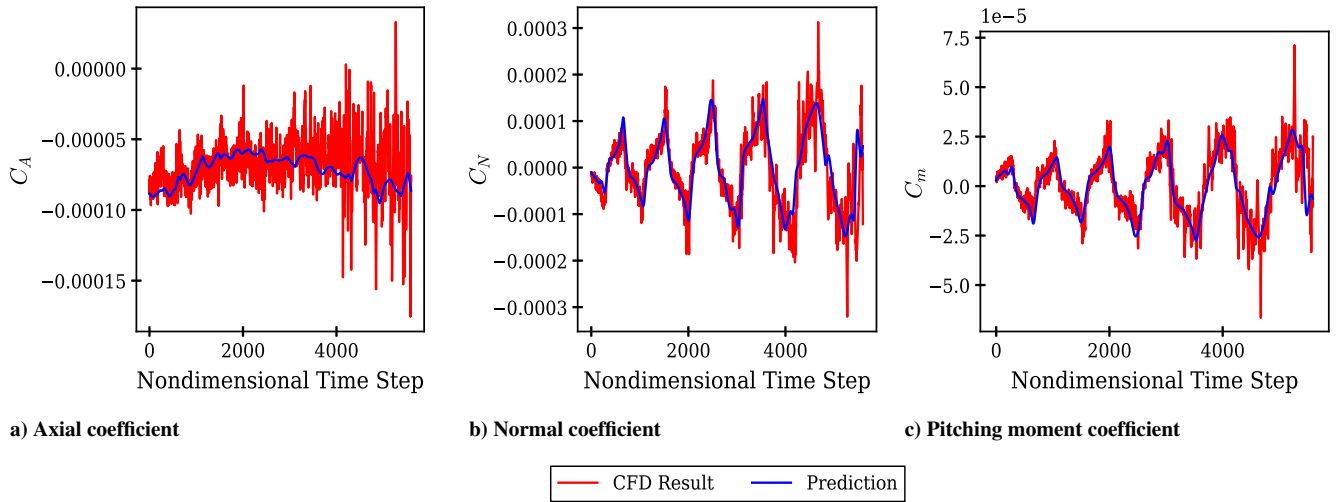


Fig. 27 Actual vs predicted C_F contributions to aerodynamic coefficients across trajectory 113—backshell.

common trend in the C_F ROMs. Future work can explore the root cause of these inaccuracies, since Fig. 16 and Fig. 17 show the C_F phenomenon is encoded into the latent spaces and POD modes during DR. The magnitude of the C_F component to the aerodynamic coefficients is minimal, continuing to raise questions about whether or not it is necessary to build C_F ROMs to capture the dynamic stability characteristics of entry vehicles.

C. Additional Validation Trajectory Evaluation

To complement the previous free-flight ROM results using trajectory 113, the following section briefly repeats the analysis using a different validation trajectory. The intent is to demonstrate that the proposed methodology remains valid when applied to a trajectory with a different dynamic behavior. Namely, trajectory 102 is used here for validation instead of trajectory 113, the former being initialized at a lower oscillation amplitude (see Table 3). As previously, the ROMs are constructed using the proposed methodology, and the validation case, trajectory 102, is held out of the training set. For conciseness, the DR results, latent coordinate predictions, and aerodynamic field reconstructions are omitted. The following section focuses instead on the resulting aerodynamic coefficients produced from the ROMs.

Figure 28 shows the heatshield C_F contributions to the aerodynamic coefficients for trajectory 102, comparing the CFD-based to the ROM-predicted aerodynamic coefficients. The results show that

the heatshield C_F ROM can accurately reconstruct the untrained aerodynamic coefficients of this low-dynamics trajectory. While some small discrepancies remain, particularly at the oscillation peaks in C_N and C_m , this suggests that the current methodology can generalize to different vehicle dynamics.

Similarly, Fig. 29 shows the backshell C_F ROM contributions to aerodynamic coefficients compared to the equivalent CFD results. The ROM predicts the mean trend, whereas the CFD-based coefficients include chaotic phenomena, which are captured in the posterior distributions of the sparse kriging model. Overall, the ROM captures the C_A mean trend well, while larger discrepancies exist in C_N and C_m at the beginning of the trajectory. Since these errors occur in the early states of the trajectory, they could propagate into larger changes in the vehicle's final state. Experiments would be required to evaluate how impactful these discrepancies could be on a simulated vehicle flight.

Figure 30 continues with the validation results of the aerodynamic coefficients computed with heatshield C_F . The discrepancy between the ROM-predicted and CFD-computed coefficients is the largest so far in the C_A results, while the difference is much smaller for C_N and C_m . The C_A trend shown in Fig. 16 shows much more oscillation compared to the smooth trend seen in 30. These differences could be causing the errors in the ROM.

Finally, Fig. 31 shows the backshell C_F contributions to aerodynamic coefficients for trajectory 102. Similar to the previous C_F

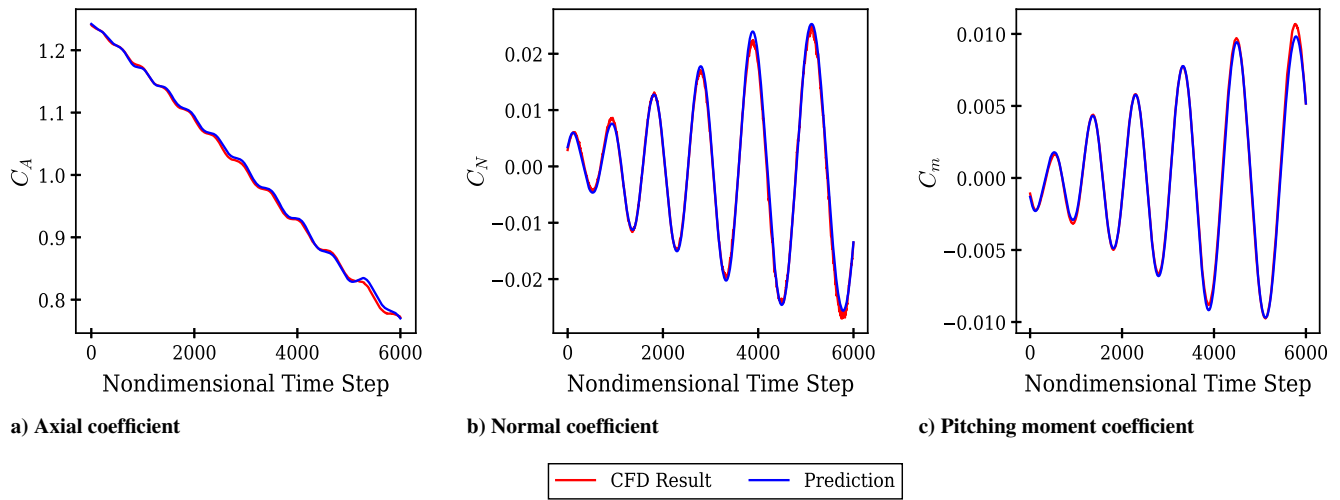


Fig. 28 Actual vs predicted C_p contributions to aerodynamic coefficients across trajectory 102—heatshield.

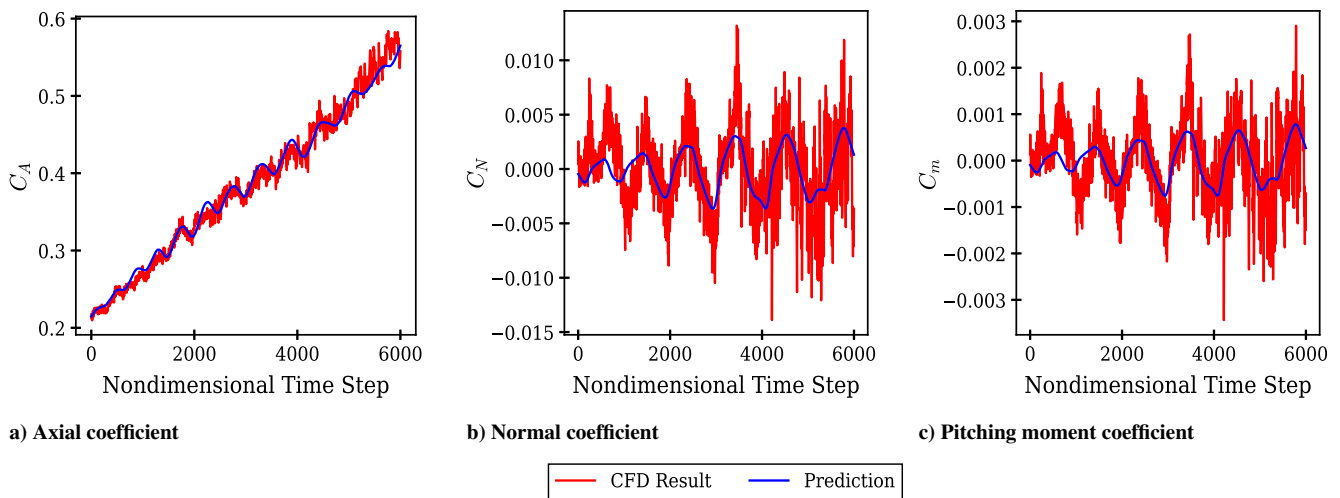


Fig. 29 Actual vs predicted C_p contributions to aerodynamic coefficients across trajectory 102—backshell.

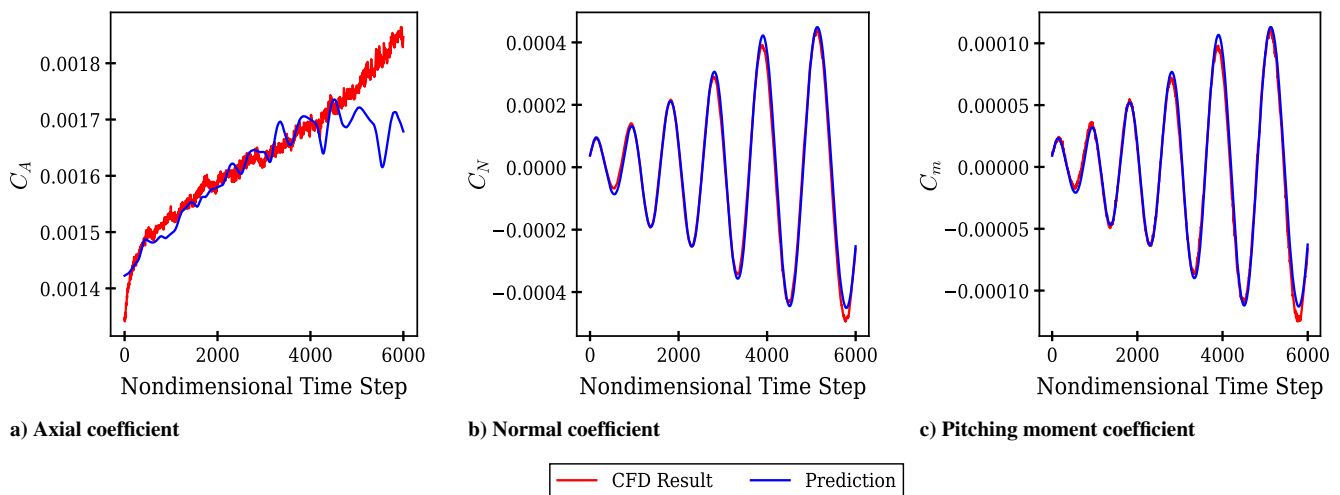


Fig. 30 Actual vs predicted C_f contributions to aerodynamic coefficients across trajectory 102—heatshield.

results, the trends for C_N and C_m exhibit a larger discrepancy at the beginning of the trajectory. The error trends in C_A also appear similar to those validated against trajectory 113, indicating that the root cause of the error may be related to characteristics of that particular field. However, it is important to consider how these small

errors impact the overall vehicle dynamics when considering all the fields and their respective contributions. Since the backshell C_F aerodynamic coefficient contributions are orders of magnitude smaller than others, one can still achieve accurate vehicle dynamic predictions despite their marginally larger error.

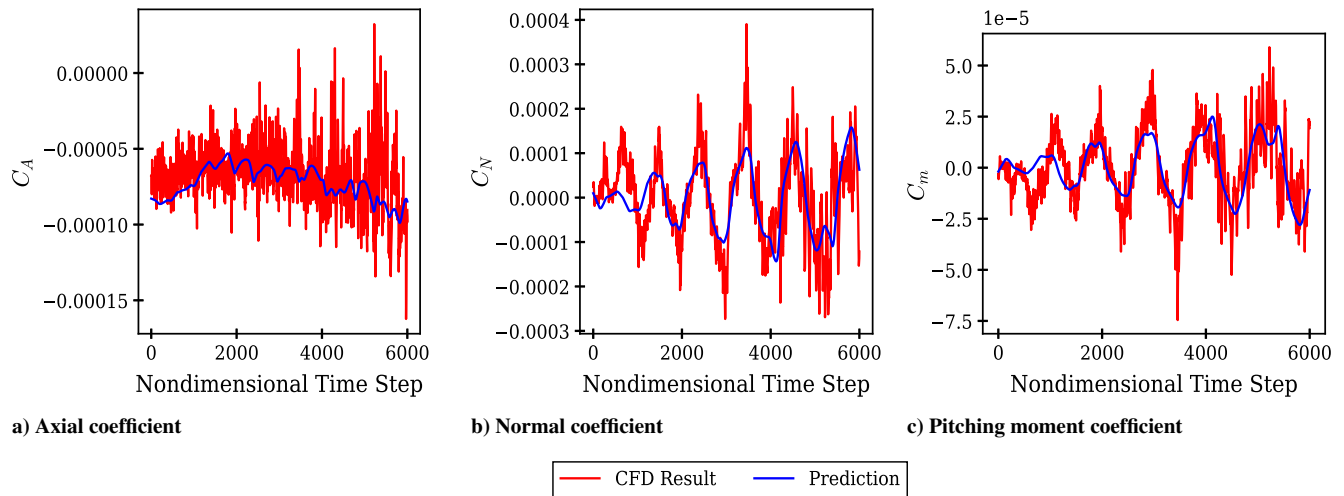


Fig. 31 Actual vs predicted C_F contributions to aerodynamic coefficients across trajectory 102—backshell.

V. Conclusions

The results of this research support the use of ROMs to quantify the dynamic aerodynamic characteristics of blunt-body entry vehicles. It expands on earlier efforts from Dean et al. [15] by applying the ROM methodology to CFD-in-the-loop flight simulation data, as well as the prediction of the skin-friction field. Another contribution of the current work is the use of sparse kriging regression to efficiently train the ROMs on a dataset containing hundreds of thousands of snapshots. Overall, the results continue to show that a linear subspace is sufficient to capture the dominant characteristics of the aerodynamic fields, including the C_F data. Analysis of the C_P backshell latent space over time identified a lagged pressure phenomenon being captured in the latent space similar to observed trends in experimental testing of entry vehicles [59]. Validations have shown that the sparse kriging regression could accurately predict the mean aerodynamic trends in C_P and C_F data for both the heatshield and backshell regions. Notably, this indicates that free-flight data is easier to predict using a ROM than forced rotational oscillation data of Dean et al. [15]. Larger errors were seen when validating against trajectory 102 than trajectory 113, indicating areas for improvement in the ROM methodology. The accuracy of the ROM predictions shows the potential of this methodology to bypass dynamic stability coefficients, predicting the combined static-dynamic aerodynamic fields throughout the trajectory. Additionally, the resulting aerodynamic coefficients of the explored ROMs provided insights into the contributions of each data region. The C_P data appears to dominate the total aerodynamic coefficients of the entry capsule, with C_F contributions being several orders of magnitude smaller. From this observation, one could argue that the prediction of skin friction is not essential to accurately capture the dynamic stability characteristics of a blunt-body entry vehicle.

There are several avenues for future work based on this research. For instance, one could explore additional validation trajectories to further demonstrate the ROM robustness and identify additional weaknesses of this formulation. More analysis should be conducted on the latent space trends and resulting POD modes to identify if the extracted characteristics have physical meaning. This would likely require experimental testing to validate the surface distributions of the CFD simulation. Advances in pressure-sensitive paint (PSP) testing may make this possible for the C_P distributions, while it may prove more difficult for the C_F distributions. It may be possible to validate the C_F directions by tufting experimental models and using cameras to track the tuft directions during dynamic testing. Additionally, a stable and unstable definition from the ROM should be explored. Traditionally, stability coefficients are used to determine if the vehicle is stable or unstable. Since the ROMs do not use these traditional coefficients, other methods for defining a stable or unstable vehicle should be explored. It is possible that exploring the wake structures of the vehicle may provide more insight into

defining stability within the ROM, as well as help identify characteristics of the POD modes.

While ROMs continue to show promise in accurately predicting the aerodynamics of entry vehicles, their use in the dynamic simulation of an entry vehicle in flight has yet to be shown. Up to this point, the ROMs have been evaluated by providing the explicit vehicle states of a validation trajectory, evaluating only the accuracy of the aerodynamic predictions. In the context of a dynamic simulation, the ROM aerodynamic predictions would impact the resulting dynamics of the entry vehicle over time. The small prediction errors shown in the backshell C_P or C_F latent spaces may amount to a larger impact on the resulting dynamics of the capsule. As such, efforts are underway to embed the ROM predictions into a trajectory simulation, allowing the aerodynamic predictions to impact the resulting dynamics. Additionally, this capability could enable the evaluation of the C_F contribution to the dynamics and determine if this field could be omitted in practice.

Acknowledgments

This work was funded by Early Stage Innovations grant 80NSSC23K0229 from NASA's Space Technology Research Grants Program. Additionally, the computing resources and technical support enabling this work were provided in part by the Partnership for an Advanced Computing Environment at the Georgia Institute of Technology. The CFD-in-the-loop flight simulation data, which provided the training data for the ROMs, was generated by Miguel Walters and Brenton Willier, both at the Aerospace Systems Design Laboratory at the Georgia Institute of Technology. Finally, the authors would like to thank Eli Shellabarger, Zachary Ernst, Alexandra Hickey, and Ken Decker for their support of this work.

References

- [1] Schoenenberger, M., and Queen, E., "Limit Cycle Analysis Applied to the Oscillations of Decelerating Blunt-Body Entry Vehicles," *NATO RTO Symposium AVT-152 on Limit-Cycle Oscillations and Other Amplitude-Limited, Self-Excited Vibrations*, RTO-MP-AVT-152, May 2008, <https://ntrs.nasa.gov/citations/20080018707>.
- [2] Owens, D. B., and Aubuchon, V., "Overview of Orion Crew Module and Launch Abort Vehicle Dynamic Stability (Invited)," *29th AIAA Applied Aerodynamics Conference*, AIAA Paper 2011-3504, June 2011. <https://doi.org/10.2514/6.2011-3504>
- [3] Moseley, W. C., Jr., Moore, R. H., Jr., and Hughes, J. E., "Stability Characteristics of the Apollo Command Module," NASA TN D-3890, Hampton, VA, 1967.
- [4] Chapman, G. T., and Yates, L. A., "Limit Cycle Analysis of Planetary Probes," *37th Aerospace Sciences Meeting and Exhibit*, AIAA Paper 1999-1022, 1999. <https://doi.org/10.2514/6.1999-1022>
- [5] Hickey, A. M., Ernst, Z., Robertson, B. E., and Mavris, D. N., "Demonstration of a Coupled CFD-RBD-FCS Framework on Mars Science

- Laboratory Vehicle," *AIAA SCITECH 2023 Forum*, AIAA Paper 2023-1167, 2023.
<https://doi.org/10.2514/6.2023-1167>
- [6] Slotnick, J., Khodadoust, A., Alonso, J., and Darmofal, D., "CFD Vision 2030 Study: A Path to Revolutionary Computational Aerodynamics," NASA CR 2014-218178, Hampton, VA, 2014.
 - [7] Ernst, Z. J., Robertson, B. E., and Mavris, D., "Utilizing Artificial Neural Networks for Entry Vehicle Aerodynamic Characterization," *Journal of Spacecraft and Rockets*, Vol. 61, No. 6, 2024, pp. 1–11.
<https://doi.org/10.2514/1.A35737>
 - [8] Murman, S. M., "Dynamic Viscous Simulations of Atmospheric-Entry Capsules," *AIAA Applied Aerodynamics Conference*, AIAA Paper 2008-6911, 2008.
<https://doi.org/10.2514/6.2008-6911>
 - [9] Stern, E., Gidzak, V., and Candler, G., "Estimation of Dynamic Stability Coefficients for Aerodynamic Decelerators Using CFD," *30th AIAA Applied Aerodynamics Conference*, AIAA Paper 2012-3225, 2012.
<https://doi.org/10.2514/6.2012-3225>
 - [10] Teramoto, S., Hiraki, K., and Fujii, K., "Numerical Analysis of Dynamic Stability of a Reentry Capsule at Transonic Speeds," *AIAA Journal*, Vol. 39, No. 4, 2001, pp. 646–653.
<https://doi.org/10.2514/2.1357>
 - [11] Brock, J. M., and Kazemba, C. D., "Dynamic Stability Methodologies and Capabilities," *AIAA SCITECH 2023 Forum*, AIAA Paper 2023-1335, 2023.
<https://doi.org/10.2514/6.2023-1335>
 - [12] Kazemba, C. D., Braun, R. D., Clark, I. G., and Schoenenberger, M., "Survey of Blunt-Body Supersonic Dynamic Stability," *Journal of Spacecraft and Rockets*, Vol. 54, No. 1, 2017, pp. 109–127.
<https://doi.org/10.2514/1.A33552>
 - [13] McKown, Q. E., Kazemba, C., Stern, E., and Brock, J., "Attitude Reconstruction of Free-Flight CFD Generated Trajectories Using Non-Linear Pitch Damping Coefficient Curves," *AIAA SCITECH 2022 Forum*, AIAA Paper 2022-1169, 2022.
<https://doi.org/10.2514/6.2022-1169>
 - [14] Anderson, J. D., *Fundamentals of Aerodynamics*, 3rd ed., McGraw Hill, New York, 1984, p. 15.
 - [15] Dean, H. V., Decker, K., Robertson, B. E., and Mavris, D. N., "Reduced-Order Modeling Approach to Blunt-Body Aerodynamic Modeling," *Journal of Spacecraft and Rockets*, Vol. 62, No. 4, July 2025.
<https://doi.org/10.2514/1.A36095>
 - [16] Decker, K., Iyengar, N., Perron, C., Rajaram, D., and Mavris, D., "Nonlinear Multi-Fidelity Reduced Order Modeling Method Using Manifold Alignment," *AIAA Aviation 2021 Forum*, AIAA Paper 2021-3050, 2021.
<https://doi.org/10.2514/6.2021-3050>
 - [17] Bui-Thanh, T., Damodaran, M., and Willcox, K., "Proper Orthogonal Decomposition Extensions for Parametric Applications in Compressible Aerodynamics," *21st AIAA Applied Aerodynamics Conference*, AIAA Paper 2003-4213, 2003.
<https://doi.org/10.2514/6.2003-4213>
 - [18] Rajaram, D., Gautier, R., Perron, C., Fischer, O. J., and Mavris, D. N., "Non-Intrusive Parametric Reduced Order Models with High-Dimensional Inputs via Gradient-Free Active Subspace," *AIAA Aviation 2020 Forum*, AIAA Paper 2020-3184, 2020.
<https://doi.org/10.2514/6.2020-3184>
 - [19] Rajaram, D., Perron, C., Puranik, T. G., and Mavris, D. N., "Randomized Algorithms for Non-Intrusive Parametric Reduced Order Modeling," *AIAA Journal*, Vol. 58, No. 12, 2020, pp. 5389–5407.
<https://doi.org/10.2514/1.J059616>
 - [20] Decker, K. H., Schwartz, H. D., and Mavris, D. N., "Dimensionality Reduction Techniques Applied to the Design of Hypersonic Aerial Systems," *AIAA Aviation 2020 Forum*, AIAA Paper 2020-3003, 2020.
<https://doi.org/10.2514/6.2020-3003>
 - [21] Mignolet, M. P., Przekop, A., Rizzi, S. A., and Spottswood, S. M., "A Review of Indirect/Non-Intrusive Reduced Order Modeling of Non-linear Geometric Structures," *Journal of Sound and Vibration*, Vol. 332, No. 10, 2013, pp. 2437–2460.
<https://doi.org/10.1016/j.jsv.2012.10.017>
 - [22] Perron, C., Rajaram, D., and Mavris, D. N., "Multi-Fidelity Non-Intrusive Reduced-Order Modelling Based on Manifold Alignment," *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, Vol. 477, No. 2253, 2021, Paper 20210495.
<https://doi.org/10.1098/rspa.2021.0495>
 - [23] Iyengar, N., Rajaram, D., Decker, K., Perron, C., and Mavris, D., "Nonlinear Reduced Order Modeling Using Domain Decomposition," *AIAA SciTech Forum 2022*, AIAA Paper 2022-1250, 2022.
<https://doi.org/10.2514/6.2022-1250>
 - [24] Dol, S. S., "Proper Orthogonal Decomposition Analysis of Vortex Shedding Behind a Rotating Circular Cylinder," *EPJ Web of Conferences*, Vol. 114, 2016, pp. 1–8.
<https://doi.org/10.1051/epjconf/201611402019>
 - [25] Berkooz, G., Holmes, P., and Lumley, J. L., "The Proper Orthogonal Decomposition in the Analysis of Turbulent Flows," *Annual Reviews*, Vol. 25, No. 1, 1993, pp. 539–575.
<https://doi.org/10.1146/annurev.fl.25.010193.002543>
 - [26] Johansson, P. B., George, W. K., and Woodward, S. H., "Proper Orthogonal Decomposition of an Axisymmetric Turbulent Wake Behind a Disk," *Physics of Fluids*, Vol. 14, No. 7, 2002, pp. 2508–2514.
<https://doi.org/10.1063/1.1476301>
 - [27] Bui-Thanh, T., Willcox, K., and Ghattas, O., "Parametric Reduced-Order Models for Probabilistic Analysis of Unsteady Aerodynamic Applications," *AIAA Journal*, Vol. 46, No. 10, 2008, pp. 2520–2529.
<https://doi.org/10.2514/1.35850>
 - [28] Peters, N. J., Wissink, A., and Ekaterinaris, J., "On the Construction of a Mode Based Reduced Order Model for a Moving Store," *Aerospace Science and Technology*, Vol. 123, 2022, Paper 107484.
<https://doi.org/10.1016/j.ast.2022.107484>
 - [29] Xiao, D., Fang, F., Pain, C. C., and Navon, I. M., "A Parameterized Non-Intrusive Reduced Order Model and Error Analysis for General Time-Dependent Nonlinear Partial Differential Equations and Its Applications," *Computer Methods in Applied Mechanics and Engineering*, Vol. 317, April 2017, pp. 868–889.
<https://doi.org/10.1016/j.cma.2016.12.033>
 - [30] Benner, P., Gugercin, S., and Willcox, K., "A Survey of Projection-Based Model Reduction Methods for Parametric Dynamical Systems," *SIAM Review*, Vol. 57, No. 4, 2015, pp. 483–531.
<https://doi.org/10.1137/130932715>
 - [31] Ernst, Z., Drosendahl, M., Robertson, B. E., and Mavris, D. N., "Development of a Trajectory-Centric CFD-RBD Framework for Advanced Multidisciplinary/Multiphysics Simulation," *AIAA SCITECH 2022 Forum*, AIAA Paper 2022-1793, 2022.
<https://doi.org/10.2514/6.2022-1793>
 - [32] Ernst, Z., Robertson, B. E., and Mavris, D. N., "Construction of Entry Vehicle Aerodynamic Surrogates from CFD-in-the-Loop Flight Simulations," *AIAA SCITECH 2023 Forum*, AIAA Paper 2023-116, 2023.
<https://doi.org/10.2514/6.2023-1166>
 - [33] Ernst, Z. J., Hickey, A. M., Robertson, B. E., and Mavris, D., "The Effect of Roll Rate on Simulated Entry Vehicle Ballistic Range Tests," *Journal of Spacecraft and Rockets*, Vol. 60, No. 1, 2023, pp. 261–272.
<https://doi.org/10.2514/1.A35131>
 - [34] Carlson, J.-R., Dutta, S., Ernst, Z., Jacobson, K., Jones, W. T., Korzun, A. M., Lugo, R. A., Nastac, G., Nielsen, E. J., Walden, A., et al., "High-Fidelity Simulations of Human-Scale Mars Lander Descent Trajectories," *AIAA Aviation 2023 Forum*, AIAA Paper 2023-3693, 2023.
<https://doi.org/10.2514/6.2023-3693>
 - [35] Nastac, G., Ernst, Z., Hickey, A., Walden, A., Jacobson, K., Jones, W., Nielsen, E. J., Diskin, B., Wang, L., Korzun, A., et al., "Closed-Loop Simulations of Human-Scale Mars Lander Descent Trajectories on Frontier," *AIAA Aviation Forum and Ascend*, AIAA Paper 2024-3535, 2024.
<https://doi.org/10.2514/6.2024-3535>
 - [36] Dean, H. V., Perron, C., Robertson, B. E., and Mavris, D. N., "Reduced Order Modeling of an Entry Capsule from Simulated Ballistic Range Trajectories," *AIAA SCITECH. 2025 Forum*, AIAA Paper 2024-1914, 2025.
<https://doi.org/10.2514/6.2024-1914>
 - [37] Taira, K., Brunton, S. L., Dawson, S. T., Rowley, C. W., Colonius, T., McKeon, B. J., Schmidt, O. T., Gordeyev, S., Theofilis, V., and Ukeiley, L. S., "Modal Analysis of Fluid Flows: An Overview," *AIAA Journal*, Vol. 55, No. 12, 2017, pp. 4013–4041.
<https://doi.org/10.2514/1.J056060>
 - [38] Tropea, C., Yarin, A., and Foss, J., *Springer Handbook of Experimental Fluid Mechanics*, Springer-Verlag, Berlin, 2007, p. 237.
 - [39] Maulik, R., and Mengaldo, G., "PyParSVD: A Streaming, Distributed and Randomized Singular-Value-Decomposition Library," *CoRR*, Vol. abs/2108.08845, 2021, <https://arxiv.org/abs/2108.08845>.
 - [40] Iyengar, N., Rajaram, D., Decker, K., and Mavris, D. N., "Uncertainty Propagation in High-Dimensional Fields Using Non-Intrusive Reduced Order Modeling and Polynomial Chaos," *AIAA SCITECH 2023 Forum*, AIAA Paper 2023-1686, 2023.
<https://doi.org/10.2514/6.2023-1686>
 - [41] Franz, T., Zimmermann, R., Görtz, S., and Karcher, N., "Interpolation-Based Reduced-Order Modelling for Steady Transonic Flows via Manifold Learning," *International Journal of Computational Fluid Dynamics*, Vol. 28, Nos. 3–4, 2014, pp. 106–121.
<https://doi.org/10.1080/10618562.2014.918695>

- [42] Aversano, G., D'Alessio, G., Coussement, A., Contino, F., and Parente, A., "Combination of Polynomial Chaos and Kriging for Reduced-Order Model of Reacting Flow Applications," *Results in Engineering*, Vol. 10, June 2021, Paper 100223.
<https://doi.org/10.1016/j.rineng.2021.100223>
- [43] Rasmussen, C. E., and Williams, C. K. I., *Gaussian Processes for Machine Learning*, MIT Press, Cambridge, MA, 2005.
<https://doi.org/10.7551/mitpress/3206.001.0001>
- [44] Forrester, A. I., and Keane, A. J., "Recent Advances in Surrogate-Based Optimization," *Progress in Aerospace Science*, Vol. 45, Nos. 1–3, 1 2009, pp. 50–79.
<https://doi.org/10.1016/j.paerosci.2008.11.001>
- [45] Liu, H., Ong, Y.-S., Shen, X., and Cai, J., "When Gaussian Process Meets Big Data: A Review of Scalable GPs," *IEEE Transactions on Neural Networks and Learning Systems*, Vol. 31, No. 11, 2020, pp. 4405–4423.
<https://doi.org/10.1109/TNNLS.2019.2957109>
- [46] Seeger, M. W., Williams, C. K., and Lawrence, N. D., "Fast Forward Selection to Speed Up Sparse Gaussian Process Regression," *International Workshop on Artificial Intelligence and Statistics*, PML, 2003, pp. 254–261.
- [47] Snelson, E., and Ghahramani, Z., "Sparse Gaussian Processes Using Pseudo-Inputs," *Advances in Neural Information Processing Systems*, Vol. 18, MIT Press, Cambridge, MA, 2005, https://proceedings.neurips.cc/paper_files/paper/2005/file/4491777b1aa8b5b32c2e8666db1a495-Paper.pdf.
- [48] Bui, T. D., Yan, J., and Turner, R. E., "A Unifying Framework for Gaussian Process Pseudo-Point Approximations Using Power Expectation Propagation," *Journal of Machine Learning Research*, Vol. 18, No. 104, 2017, pp. 1–72.
<https://doi.org/10.48550/arXiv.1605.07066>
- [49] Titsias, M. K., "Variational Learning of Inducing Variables in Sparse Gaussian Processes," *Twelfth International Conference on Artificial Intelligence and Statistics*, PMLR, 2009, <http://proceedings.mlr.press/v5/titsias09a/titsias09a.pdf>.
- [50] Bauer, M., Van der Wilk, M., and Rasmussen, C. E., "Understanding Probabilistic Sparse Gaussian Process Approximations," *Advances in Neural Information Processing Systems*, Vol. 29, 2016.
<https://doi.org/10.48550/arXiv.1606.04820>
- [51] Bradbury, J., Frostig, R., Hawkins, P., Johnson, M. J., Leary, C., Maclaurin, D., Necula, G., Paszke, A., VanderPlas, J., Wanderman-Milne, S., et al., "JAX: Composable Transformations of Python + NumPy Programs," 2018, <http://github.com/google/jax>.
- [52] Cheatwood, F., Winchenbach, G., Hathaway, W., and Chapman, G., "Dynamic Stability Testing of the Genesis Sample Return Capsule," *38th Aerospace Sciences Meeting and Exhibit*, AIAA Paper 2000-1009, 2000.
<https://doi.org/10.2514/6.2000-1009>
- [53] Prince, J. L., Desai, P. N., Queen, E. M., and Grover, M. R., "Mars Phoenix Entry, Descent, and Landing Simulation Design and Modeling Analysis," *Journal of Spacecraft and Rockets*, Vol. 48, No. 5, 2011, pp. 756–764.
<https://doi.org/10.2514/1.46561>
- [54] Schuster, D. M., "CFD, Grand Challenge: CFD-in-the-Loop Monte Carlo Flight Simulation for Space Vehicle Design," *AIAA Scitech 2021 Forum*, AIAA Paper 2021-0957, 2030, pp. 1–12.
<https://doi.org/10.2514/6.2021-0957>
- [55] Schoenenberger, M., Cheatwood, F. M., and Desai, P., "Static Aerodynamics of the Mars Exploration Rover Entry Capsule," *43rd AIAA Aerospace Sciences Meeting and Exhibit*, AIAA Paper 2005-56, 2005, 13,139, 13,150.
<https://doi.org/10.2514/6.2005-56>
- [56] Desai, P. N., and Cheatwood, F. M., "Entry Dispersion Analysis for the Genesis Sample Return Capsule," *Journal of Spacecraft and Rockets*, Vol. 38, No. 3, 2001, pp. 345–350.
<https://doi.org/10.2514/2.3707>
- [57] Anderson, W. K., Biedron, R. T., Carlson, J.-R., Derlaga, J. M., Druyor, C. T., Gnoffo, P. A., Hammond, D. P., Jacobson, K. E., Jones, W. T., Kleb, B., et al., "FUN3D Manual: 14.0.1," NASA TM 20230004211, Hampton, VA, April 2023.
- [58] Noack, B. R., Stankiewicz, W., Morzyński, M., and Schmid, P. J., "Recursive Dynamic Mode Decomposition of Transient and Post-Transient Wake Flows," *Journal of Fluid Mechanics*, Vol. 809, Dec. 2016, pp. 843–872.
<https://doi.org/10.1017/jfm.2016.678>
- [59] Kazemba, C. D., Braun, R. D., Schoenenberger, M., and Clark, I. G., "Dynamic Stability Analysis of Blunt-Body Entry Vehicles Using Time-Lagged Aftbody Pitching Moments," *Journal of Spacecraft and Rockets*, Vol. 52, No. 2, 2015, pp. 393–403.
<https://doi.org/10.2514/1.A32894>
- [60] Brockwell, P. J., and Davis, R. A., *Time Series: Theory and Methods*, 2nd ed., Springer-Verlag, New York, 1991.

K. T. Edquist
Associate Editor