Non-intrusive model reduction of static parametric non-linear systems and application to global optimization and uncertainty quantification

S. Ashwin Renganathan*1 | Dimitri N. Mavris2

1Postdoctoral Fellow, School of Aerospace Engineering, Georgia Institute of Technology, Georgia, United States of America
2Regents Professor, School of Aerospace Engineering, Georgia Institute of Technology, Georgia, United States of America

Correspondence
*S. Ashwin Renganathan. Email: ashwinsr@gatech.edu

Present Address
270 Ferst Drive NW, Atlanta, GA 30332

Summary
A methodology for non-intrusive parametric model reduction is presented as an extension of the basic method introduced by Renganathan et. al. (2018). Specifically, we extend the method for static systems with parametric geometry (that deforms the mesh), in addition to parametric boundary conditions. The main idea is to first perform a transformation on the governing equations such that it is lifted to a higher dimensional but linear under-determined system. This enables one to extract the system matrices easily compared to that of the original non-linear system. The under-determined system is closed with a set of unique non-linear constraints upon which the model reduction is performed. The methodology is validated on the subsonic and transonic inviscid flow past the NACA0012 and the RAE2822 airfoils. We further demonstrate the utility of the approach by applying it to two common problems in the many-query context namely, gradient-free global optimization and parametric uncertainty quantification with Monte-Carlo sampling. Overall, the methodology is shown to achieve accuracy up to 5% and computational speed-up of 2-3 orders of magnitude as that of the full-order model.

KEYWORDS:
Model Order Reduction, surrogate modeling, non-intrusive methods, global optimization, uncertainty quantification

1 INTRODUCTION

In the design of complex aerospace engineering systems such as aircraft, rotorcraft and launch vehicles, the advent of Computer Aided Engineering (CAE) and high performance computing has significantly contributed in reducing the time and cost involved in the design process. Specifically, high-fidelity mathematical models such as Computational Fluid Dynamics (CFD) has to a good extent substituted expensive physical testing with computer simulations. However when it comes to decision making in the design of aerospace systems, such high-fidelity models have to be queried several thousands of times, which is not practical since each query could take few days to weeks to compute even on a supercomputer. Specifically, we focus on real-time decision making via solving problems involving global optimization and uncertainty quantification.

Partial Differential Equations (PDE) based models such as CFD, typically require an iterative solution whose computational cost scales in polynomial time with the number of degrees of freedom in the spatial discretization. Practical problems of

Submitted to the editors
aerospace interest, typically involve $\Theta(10^6)$ degrees of freedom, and therefore their application in the many-query context is not feasible. Therefore a surrogate model that would retain most of the accuracy of the high-fidelity model while only requiring a fraction of its computational cost is needed. Such a model would enable reliable, real-time decision making thereby offering a necessary paradigm shift in the existing process in aerospace design.

We specifically focus on Reduced Order Modeling (ROM) or Model Order Reduction (MOR) which retain the underlying physical characteristics of the high-fidelity model (henceforth referred as Full Order Model) by projecting the actual governing equations onto a suitably chosen low-dimensional subspace, for which Proper Orthogonal Decomposition (POD) is a common technique. ROMs are quite popular for this reason, however are limited to situations only where there is access to the governing equations of the Full Order Model (FOM). In situations where the governing equations are available as a black-box (such as in commercial codes), projection-based ROM is not feasible and hence a non-intrusive technique is necessary. In such situations, the common approach taken is to circumvent the projection step all together and use the POD to directly approximate the state whose generalized coordinates are interpolated in the time/parameter space (see [3,4,5,6]). Such an approach is effective in the sense that it is purely data-driven and is more general in its applicability. However they do not guarantee that the resulting ROM still satisfies the actual governing equations. There has also been work done in data-driven discovery of the governing equations from black-box codes which can then be used for model reduction. [7] showed how the reduced operators of the discretized governing equations can be inferred from the state trajectory, boundary and initial conditions. However, in this work, we are focused on static parametric systems where there is no trajectory data and further the boundary conditions might not be explicitly available. In an earlier work, Renganathan et al (2018) showed that by lifting the system to a higher dimension via the Koopman theory, a linear but under-determined system can be obtained which can then be closed with a set of non-linear problem-specific constraints. They further showed that with such a technique the discrete linear operator can be extracted by discretizing the linear differential terms via a method such as finite volume method, at a cost that scales linearly with the grid size, $N$. However, the method was restricted only to parameters in the boundary conditions. Here, we further extend that method to apply towards systems with parametric geometry, where the grid varies at every parameter snapshot. We show the effectiveness of the approach by generating a database of ROMs for a pre-determined set of snapshots and later interpolating between the ROMs.

The rest of the paper is organized as follows. The methodology is first outlined. Following that the model validation is shown for subsonic and transonic inviscid flow past airfoils, to demonstrate its predictive capability. The methodology is then applied to a global optimization and uncertainty quantification problem to demonstrate its computational efficiency. The conclusion section summarizes the main findings and outlines some directions for future work. Finally, the appendices provide further details on the methodology and results sections.

## 2 | METHODOLOGY

The Koopman theory forms the basis of the present approach where a linear representation of the non-linear system is obtained. The finite volume method is the tool used to discretize linear differential terms as they are well suited for unstructured computational grids and are almost the standard in commercial CFD applications. As mentioned before, this approach depends on generating a database of ROMs corresponding to a pre-determined set of parameter snapshots which are then interpolated for new realizations of the parameters outside of the training set. The ROM interpolation draws from differential geometry in order to address the manifold-embedding of the ROM system matrices. Such an approach is essential in ensuring the fundamental properties of the system matrices are retained post-interpolation. An outline of the methodology is provided here while the reader is referred to [10,11] for more details.

Consider a static non-linear system of the form

$$
\mathbf{N}(\mathbf{u}) = 0
$$

where $\mathbf{N}$ represents a non-linear operator acting on the state variable $\mathbf{u}$. Let $g(\mathbf{u})$ represent an observable that is a function of the state variable, $\mathbf{u}$. We state that

$$
\mathbf{N}(\mathbf{u}) \to \mathbf{L}[g(\mathbf{u})]
$$

where, $\mathbf{L}$ is a linear operator acting on the observable. Discretizing the above equation, we get

$$
\mathbf{L}[g(\mathbf{u})] \approx \mathbf{A}g(\mathbf{u}) + \mathbf{b}_u = 0
$$
Where, $b_a$ represents a vector that arises due to the discretization of the boundary conditions, lumps the source terms if present and is also the RHS of the FOM. Note again that since each non-linear term is transformed into an observable, the term $b_a$ in Equation 3 no longer depends on the state $u$ and hence Equation 3 is a linear (but under-determined) system. For uniqueness of the solution of 3, certain constraints are added as discussed in section 2.1. Setting $g(u) \rightarrow y$ and $-b_a \rightarrow f$, leads to the transformed linear system

$$Ay = f$$

We intend to develop the ROM for Equation 4 and hence the snapshots are collected in the observable ($y$) space. In the present work, we differ from the traditional Koopman-DMD used for dynamical systems by not using any trajectory data (we address static systems). Secondly, the linear operator of the transformed equation is directly approximated by discretizing the linear differential terms on arbitrary unstructured grids via finite volume method. This operator, applied to a snapshot of the observables itself gives the RHS of the governing equations that lump the non-linear terms, boundary conditions and source terms if present. This way, the present work obtains a linear representation of the governing equations, which is comparatively easier to work with. However, transforming the state-space to observable-space to obtain a linear system does not necessarily form a closed linear system that is consistent with the original non-linear system. And therefore lastly, this work differs in that the linear system is closed with a set of non-linear constraints that establish consistency between the original state variables and the observables. In essence, the state-observable transformation to lift the original system is the only step that is drawn from the Koopman theory. Finally, the model reduction is performed on the transformed equations (the right hand side of Figure 1). This is explained in the following sub-section.

FIGURE 1 Graphical depiction of the present methodology. The original non-linear static system is transformed to an under-determined linear system with closure.
2.1 Model Order Reduction

The observables in Equation 4 are $y = [y_1^T, ..., y_O^T]^T$, where $O$ is the total number of linear and non-linear terms in the FOM and also the total number of observables and $y_i \in \mathbb{R}^N$. For a FOM that is a system of $S$ coupled PDEs, note that $O \geq S$ always and $O > S$ for a non-linear system. Therefore the observables can be written

$$y = [y_1^T, ..., y_S^T, y_{S+1}^T, ..., y_O^T]^T$$

As evident from the previous equation, a side effect of representing the FOM in terms of observables is that it can lift the original system to a higher dimension. In this work, we add algebraic equations that establish the non-linear consistency relationship between certain observables and the rest in order to provide closure to the under-determined system. These constraints are of the form

$$h_i(y) := y_{S+i} - f(y_1, ..., y_S) = 0, \ i = 1, ..., O - S$$

Equation 6 along with the transformed FOM in Equation 4 together form a closed system upon which model reduction is performed. Denoting $\Phi_i \in \mathbb{R}^{N \times k_i}$ to be the reduced POD bases extracted from snapshots of observable $y_i$, the trial basis matrix is defined as

$$\Phi = \begin{bmatrix} \Phi_1 & \cdots & \Phi_O \end{bmatrix} \in \mathbb{R}^{ON \times k}$$

where $k = k_1 + \cdots + k_O$ and $\hat{y} = \Phi_k^T y$. Note that the POD basis for each observable is separately determined from their respective snapshot matrix ($Y_i \in \mathbb{R}^{N \times M}$) that takes the following form

$$Y_i = \begin{bmatrix} \vdots & \vdots & \vdots & \vdots \\ y_i^{(1)} & y_i^{(2)} & y_i^{(3)} & \cdots \\ \vdots & \vdots & \vdots & \vdots \end{bmatrix}$$

where $y_i^{(j)}$ is the $jth$ snapshot of $y_i$, $M$ is the total number of snapshots use for model training and the POD basis vectors are obtained from the thin-svd of $Y_i$. Since the full system is non-square (due to introduction of observables), a suitable choice for the test basis for projection is $\Psi_k = A \Phi_k$. Note that this choice of the test basis is equivalent to a gakerkin projection ($\Psi_k = \Phi_k$) on the normal equations. i.e. on $A^T A y = A^T f$. Let $B = A^T A$; then the projection leads to

$$\Phi_k^T B \Phi_k \hat{y} = \Phi_k^T A^T f$$

Setting $\bar{f} = \Phi_k^T A^T f \in \mathbb{R}^k$ and $\bar{B} = \Phi_k^T B \Phi_k \in \mathbb{R}^{k \times k}$, this leads to the reduced order model

$$\bar{B} \bar{y} = \bar{f}$$

The ROM given by Equation 10 represents a $k \times k$ system which is rank-deficient since it was obtained through an outer product of two rectangular matrices and is solved along with the constraints presented in Equation 6 posed as a constrained optimization problem as shown below

$$\begin{aligned}
&\text{minimize} & & \frac{1}{2} \|\bar{B} \bar{y} - \bar{f}\|^2_2 \\
&\text{subject to} & & h(y) = 0
\end{aligned}$$

where $h(y)$ is a non-linear function (as shown in Equation 6) that represents the relationship between observables and is problem dependent. See section 3.1 for how constratins are formulated for the two-dimensional compressible Euler equations. The main hypothesis of this work is that the ROM given by Equation 11 still approximately satisfies the governing equations and this is verified in the Section 3. The optimization problem in Equation 11 needs special treatment to handle the non-linear constraint which still depends on the full state of observables, and is efficiently done using the DEIM 18; see Appendix 5.3 for details on implementation for a specific example. The ROM in Eq. 11 is solved via Sequential Quadratic Programming 32.
2.2 | ROM Interpolation

The proposed approach that leads to the ROM in the form of Eq. 10 corresponds to one parameter snapshot. Therefore, the approach generates a database of ROMs for a pre-determined set of parameter snapshots, which are later interpolated to predict the state at a new parameter. The interpolation is carried out in a manner that retains the inherent structure and properties of the matrix \( \tilde{B} \) post-interpolation. The general principle that is followed is to map the matrices to a plane that is locally tangent to the manifold in which they are originally embedded. The traditional Euclidean space interpolation (where typical vector operations are valid) is then carried out in the tangent plane after which they are mapped back to the manifold. The mapping to and from the tangent plane are carried out via logarithmic and exponential relationships as depicted in Figure 2.

![Figure 2](image)

**FIGURE 2** A graphical representation of a manifold \( \mathcal{M} \) and the embedding of parametric matrices \( \tilde{B}(\theta) \). A direct element-wise interpolation of \( \tilde{B} \) at \( \tilde{\theta} \) may not necessarily result in a matrix \( \in \mathcal{M} \), and is carried out after mapping to \( T(\mathcal{M}) \).

The matrix \( \tilde{B} \) in Equation 11 is known to be symmetric positive definite. This is because, \( \tilde{B} = \Phi^T A^T A \Phi \) and the co-variance matrix \( A^T A \) is symmetric positive semi-definite (see sec. 5.3). Additionally, multiplication by orthogonal matrix \( \Phi \) of rank \( k \) where \( k < \text{rank}(A) \) ensures \( \tilde{B} \) is symmetric positive definite. Therefore, we are seeking a tangent plane to the manifold containing the set of all symmetric positive definite matrices of a specific size to perform our matrix interpolation. Symmetric positive definite matrices of size \( n \times n \) form a special group called the \( SPD(n) \). Also, for the set of all SPD matrices \( B_n \in \mathcal{M} \), the tangent plane is the set of all symmetric matrices, \( B' \). The geodesic connecting two points \( B_1 \) and \( B_2 \) on \( \mathcal{M} \) is defined by the equation

\[
\gamma(t) = B_1^{1/2} \left( B_1^{-1/2} B_2 B_1^{-1/2} \right)^t B_1^{1/2}
\]  

(12)

Any metric \( (\mathcal{M}_f) \) defined on \( SPD(n) \) for any two matrices uses the following functional relationship

\[
\mathcal{M}_f(B_1, B_2) = B_1^{1/2} f \left( B_1^{-1/2} B_2 B_1^{-1/2} \right) B_1^{1/2}
\]  

(13)

which leads to the following results for the exponential and logarithmic mapping for \( SPD(n) \), where, \( \tilde{B}_0 \) is the anchor point and \( \tilde{B}' \) is the point whose mapping is desired.

Exponential Mapping of \( \tilde{B}' \) from tangent plane to \( \mathcal{M} \) at \( \tilde{B}_0 \in \mathcal{M} \), to \( \mathcal{M} \):

\[
\text{Exp}_{\tilde{B}_0} \tilde{B}' = \tilde{B}_0^{1/2} \left( \tilde{B}_0^{-1/2} \exp(\tilde{B}') \tilde{B}_0^{-1/2} \right) \tilde{B}_0^{1/2}
\]  

(14)

Logarithmic Mapping of \( \tilde{B} \in \mathcal{M} \) to tangent plane to \( \mathcal{M} \) at \( \tilde{B}_0 \in \mathcal{M} \):

\[
\text{Log}_{\tilde{B}_0} \tilde{B}' = \tilde{B}_0^{1/2} \log \left( \tilde{B}_0^{-1/2} \tilde{B} \tilde{B}_0^{-1/2} \right) \tilde{B}_0^{1/2}
\]  

(15)

The results presented in Equations 14 and 15 are used in this work to perform the tangent space interpolation; see for further details.

2.3 | Computational Cost

The overall computational cost is dominated by the offline phase where the model is built. Here, we aim to provide an estimate of the computational cost in terms of Floating Point OPerationS (FLOPS) necessary to build the ROM as a function of the
grid size ($N$) and snapshot size ($M$), given the snapshots (training data from high-fidelity simulations). The cost of the online phase (including ROM interpolation) is trivial comparatively and the wall-clock time is more relevant in this scenario. The off-line phase includes 4 major steps whose computational cost are summarized in Table 1; see for details. The summary of computational cost are provided in Table 1. Note that only the dominating factors of the cost are provided in the table. It can be seen that the most expensive steps of the method are the POD and the projection which scale as $\sim N^2 M$; as $N$ increases the cost of these steps increases quite rapidly. The finite volume discretization is relatively a cheaper step that scales linearly with grid size.

### 2.4 Overall Method

The overall method is summarized in Algorithm 1 and in Figure 3. Note that in the algorithm, $D$ represents the design space while $D_{\text{train}}$ represents a subset of $D$ used for snapshot generation (model training).
**Fig. 3** Flowchart representation of the overall framework.
Algorithm 1 Non-Intrusive Projection-Based Model Reduction

% OFFLINE
1: Choose $M$ snapshot locations for model training $\theta_i \in D_{train} \subset D$
2: % Solve FOM, construct snapshot matrix and extract POD bases:
3: for $i = 1$ to $O$ do
4:   $Y_i = [y_{i}^1, \ldots, y_{i}^M] \in \mathbb{R}^{N \times M}$
5:   $Y_i = V\Sigma W^T$, (thin-SVD)
6:   $\Phi_i = V(:, 1 : k_i) \in \mathbb{R}^{N \times k_i}$
7: end for
8: $\Phi_k = \text{blkdiag}\{\Phi_1, \ldots, \Phi_O\} \in \mathbb{R}^{N \times k}$ % trial basis matrix
9: for $\theta_{i}, i = 1$ to $M$ do
10:    $A \leftarrow$ Discretize Linear Operator
11:    $f \leftarrow A \times y$
12:    $\tilde{B} \leftarrow \Phi_i^T(A^TA)\Phi_i$
13:    $\tilde{f} \leftarrow \Phi_i^T A^T f$
14: end for

% ONLINE
15: % Prediction: for any $\theta' \notin D_{train}, \theta' \in D$
16: interpolate ROM
   Tangent space interpolation of $\tilde{B}(\theta')$ (Eq.14) Eq.15
   Euclidean space interpolation of $\tilde{f}(\theta')$
17: Solve ROM:
   minimize $\frac{1}{2}\|\tilde{B}\tilde{y} - \tilde{f}\|_2^2$
   s.t. $h(y) = 0$
18: Project ROM onto FOM space: $y = \Phi_k\tilde{y}$
19: Map observables back to state variables
   $u \leftarrow y$
TABLE 2 Free-stream conditions for the NACA and RAE test cases

|       | NACA       | RAE       |
|-------|------------|-----------|
| $P_{\infty}$ | 101,325 | 28,745 Pa |
| $\rho_{\infty}$ | 1.225 | 0.44 kg/m$^3$ |
| $a_{\infty}$ | 340.296 | 301.86 m/s |
| $\mu_{\infty}$ | 1.78E-5 | 1.49E-5 Pa - s |
| $M$ | 0.60 | 0.734 |
| $\alpha$ | 2.0 | 2.79 deg. |

3 | RESULTS AND DISCUSSION

We demonstrate the methodology on the inviscid flow past airfoils, for which the NACA0012 and RAE2822 are chosen as baselines at subsonic and transonic flow regimes respectively. We begin by validating the approach by comparing the ROM predictions against the FOM solution in section 3.3 following which we apply the method to the many-query problems in sections 3.4 and 3.5.

3.1 | Governing Equations

The Euler equations governing the 2D, compressible, inviscid flow past an airfoil are the governing equations on which we perform model reduction. This coupled non-linear system is solved via the commercial black-box CFD solver, STARCCM+.$^{22}$ The equations in conservation form are provided in Eq.16 below,

$$\nabla_x F + \nabla_y G = 0$$

where

$$F = \begin{bmatrix} \rho u \\ \rho u^2 + p \\ \rho u v \\ \rho u H \end{bmatrix}, \quad G = \begin{bmatrix} \rho v \\ \rho u v \\ \rho v^2 + p \\ \rho v H \end{bmatrix}$$

$$H = E + \frac{p}{\rho}$$

$$\rho E = \frac{1}{2} \rho (u^2 + v^2) + \frac{p}{\gamma - 1}$$

and $\nabla_x$ and $\nabla_y$ are the $x$ and $y$ components of the gradient operator $\nabla$ respectively. The following transformation is then performed

$$[\rho u, \rho v, \rho u v, p, \rho u^2, \rho v^2, \rho u H, \rho v H]^T \rightarrow [y_1, y_2, y_3, y_4, y_5, y_6, y_7, y_8]^T$$

from the state variables to observables, leading to the lifted model.

$$\begin{bmatrix} \nabla_x & \nabla_y \\ \nabla_y & \nabla_x \end{bmatrix} \begin{bmatrix} \nabla_x & \nabla_y \\ \nabla_y & \nabla_x \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \\ y_7 \\ y_8 \end{bmatrix} = 0$$

(17)
where in the above equation, empty spaces in the matrix denote zeros. The equation upon discretization leads to

\[
\begin{bmatrix}
G_x & G_y \\
G_y & G_x
\end{bmatrix}
\begin{bmatrix}
y_1 \\
y_2 \\
y_3 \\
y_4 \\
y_5 \\
y_6 \\
y_7 \\
y_8
\end{bmatrix}
- 
\begin{bmatrix}
b_{a1} \\
b_{a2} \\
b_{a3} \\
b_{a4} \\
b_{a5} \\
b_{a6} \\
b_{a7} \\
b_{a8}
\end{bmatrix}
= 
\begin{bmatrix}
y_1 \\
y_2 \\
y_3 \\
y_4 \\
y_5 \\
y_6 \\
y_7 \\
y_8
\end{bmatrix}
\times 
\begin{bmatrix}
G_x \\
G_y
\end{bmatrix}
\begin{bmatrix}
y \\
y
\end{bmatrix}
\]  
(18)

where, \( G_x \) and \( G_y \) represents the discrete version gradient operators \( \nabla_x \) and \( \nabla_y \) respectively and again, the empty spaces denote block matrices of zeros. The parameter-dependent \( A \) matrix is obtained directly by discretizing the linear terms \( G_x \) and \( G_y \) via finite volume method. The grid is exported in the CFD General Notation System (CGNS)\(^{33}\) for this purpose, as mentioned in Figure 3. The snapshots \( y \) are applied to the \( A \) matrix and the RHS \( f \) is extracted for each parameter. With the FOM reduced to the \( Ay = f \) form and \( A \in \mathbb{R}^{4 \times 8} \), \( y, f \in \mathbb{R}^{8} \), Equation 18 represents an under-determined system. Therefore they are closed using non-linear constraints given by Equation 19. Notice that the constraints express the relationship between the first \( S = 4 \) observables (\( y_1 \) through \( y_4 \)) and the remaining \( O - S \); (\( O = 8 \)) observables (\( y_5 \) through \( y_8 \)). It should be noted that all the observables that are in excess of the number of equations can be expressed as some function of the rest. The constraints are expressed in terms of the continuous form of the state and observable below.

\[
\begin{align*}
h_1 &= \rho u^2 - \frac{(\rho u)(\rho u)}{\rho} \equiv y_5 - \frac{y_1 y_3}{y_2} = 0 \\
h_2 &= \rho v^2 - \frac{(\rho u)(\rho u)}{\rho} \equiv y_6 - \frac{y_2 y_3}{y_1} = 0 \\
h_3 &= \rho u H - \rho u \left( E + \frac{p}{\rho} \right) \equiv y_7 - y_1 \left( E + \frac{y_1 y_3}{y_1 y_2} \right) = 0 \\
h_4 &= \rho v H - \rho v \left( E + \frac{p}{\rho} \right) \equiv y_8 - y_2 \left( E + \frac{y_2 y_3}{y_1 y_2} \right) = 0
\end{align*}
\]  
(19)

In the results to be presented, the following error metrics are used

\[
\begin{align*}
C_p \text{ Error} &= \left\| \frac{C_{p,FOM} - C_{p,ROM}}{C_{p,FOM}} \right\|_\infty \times 100 \\
C_d \text{ Error} &= \left\| \frac{C_{d,FOM} - C_{d,ROM}}{C_{d,FOM}} \right\|_\infty \times 100 \\
C_t \text{ Error} &= \left\| \frac{C_{t,FOM} - C_{t,ROM}}{C_{t,FOM}} \right\|_\infty \times 100
\end{align*}
\]  
(20)

### 3.2 Test cases and parametrization

The baseline shapes are parameterized using Class Shape Transformation (CST)\(^{23,24}\). The CST model of parametrization defines a class function \( C \) and a shape function \( S \) and the curve being parameterized is specified as their product. The main idea is that the class function serves to define a general class of geometry such as airfoils, missiles or sears-haack body, while the shape function serves to define the unique shape within a particular class of shapes (such as a NACA0012 vs RAE2822 airfoil). The class function, \( C(\psi) \) is more generally defined as

\[
C_{n_1}(\psi) := \psi^{n_1}(1 - \psi)^{n_2}
\]  
(21)
where the variable \( \psi \) represents the non-dimensional chord-wise distance. \( n_1 \) and \( n_2 \) define the specific class; for instance \( n_1 = 0.5, n_2 = 1 \) and hence \( \sqrt{\psi(1-\psi)} \) defines airfoils with rounded leading edge and a sharp trailing edge\(^{23}\). The unique shape of an airfoil is driven by the shape function, specified as follows

\[
S(\psi) = \sum_{i=0}^{n} A_i \psi^i
\]

(22)

where \( A_i \) are the coefficients which are also the shape parameters. The NACA0012 and RAE2822 are parameterized using 6 and 8 variables respectively, whose values are given by \( A_{NACA0012} \) and \( A_{RAE2822} \) where the top and bottom rows correspond to the upper and lower surfaces of the airfoils. Further details of the parameterization are provided in Appendix 5.1.

\[
A_{NACA0012} = \begin{bmatrix}
0.1689 & 0.2699 & 0.1387 \\
-0.1689 & -0.2699 & -0.1387 \\
\end{bmatrix}
\]

\[
A_{RAE2822} = \begin{bmatrix}
0.1268 & 0.4670 & 0.5834 & 0.2103 \\
-0.1268 & -0.5425 & -0.5096 & 0.0581 \\
\end{bmatrix}
\]

(23)

![Figure 4](image)

**FIGURE 4** Comparison of the CST approximation against the true curve

### 3.3 | Model Validation

**NACA0012**

The CST coefficients representing the NACA0012 baseline is perturbed \( \pm 30\% \) to generate new airfoil shapes, a sample of which is shown in Figure 5. A total of 170 such points were generated using a Latin Hypercube design, 160 of which was used in model building while the remaining was used to validate the model. The system matrix \( \tilde{B} \) is interpolated in the tangent space to the manifold of symmetric positive definite matrices, as explained in Section 2.2, while the RHS \( \tilde{f} \) is interpolated in the Euclidean space. In both cases a multi-variate polynomial in the Lagrange form is used for interpolation (see Renganathan(2018), Algorithm 1 for details).

The comparison of the ROM predicted pressure coefficient on the airfoil surface against the FOM solution is shown in Figure 6. Across all the 10 validation cases, the maximum and average errors in \( C_P \) are \( \approx 5\% \) and \( \approx 2\% \) respectively, whereas the computational speedup is \( \approx 100x \). Therefore, the effectiveness of proposed approach is well established. Similar accuracy was observed with the lift coefficient, which are tabulated in Appendix 5.2. The flow-field comparison in terms of overlaid pressure and mach number contours are shown in Figure which further re-iterate the low prediction errors observed.
FIGURE 5 Family of airfoils generated by perturbing (by ±30%) the CST coefficients of the NACA0012 baseline
FIGURE 6 $C_p$ comparison between ROM & FOM for various shape parameters for the NACA0012 test case
FIGURE 7 Comparison of Mach number and absolute pressure between ROM (dashed) & FOM (solid) with respect to shape parameters for the NACA0012 test case
RAE2822

Now we demonstrate the method in the transonic regime using the RAE2822 test case. Similar to the NACA test case, CST coefficients representing the baseline are perturbed ±30% to generate new airfoil shapes, a sample of which is shown in Figure 8. A total 160 snapshots were again used in model building. However, the freestream mach number for this case is set as $M = 0.734$ which leads to a shock whose strength and location are affected by perturbing the shape CST coefficients. The rest of the freestream conditions are summarized in Table 2.

We begin by comparing the ROM predictions against FOM solutions in terms of the coefficient of pressure distributions in Figure 9. These plots demonstrate that the ROM is able to capture the shock location with a maximum error of 5%. The $C_p$ predictions agree within an average error of 8.5%. In Figure 10, the pressure and mach number contours predicted by the ROM are compared against the true FOM solution, as overlaid contour plots. The ability of the ROM to accurately capture the flow field is emphasized from these plots. In terms of the lift and drag coefficients, the average error in the ROM predictions are 4% and 16% respectively.
FIGURE 9 Comparison of pressure coefficient $C_p$ predicted by the ROM with the true solution due to the FOM for various airfoil shapes that represent the validation cases.
FIGURE 10 Comparison of Mach number and absolute pressure between ROM (red) & FOM (black) with respect to shape parameters for the NACA0012 test case
Discussion

Validation of the ROM methodology with shape parameters is performed. Under subsonic shock-free conditions for the NACA test case, the predictions were consistently under 5% leading to very high accuracy. However, under transonic conditions with moving shocks in the RAE2822 test case, the predictive capability drops leading to an average error of about 15%. In the presence of discontinuities in the flow field such as moving shocks, a POD-based method such as the present method is unable to accurately predict shock location and strength.

Despite the limitations in the transonic regime, it has been observed that the ROM still predicts the shock within 5% chord-length variability. Further, the $C_p$ and $C_l$ are still predicted with an average error of 14% and 4% respectively. While the drag coefficient is predicted with an average error of 16%, due to its high sensitivity to discrepancy in pressure distributions, it incurs very large error in certain cases up to 50% (see Table 6). Overall, the observation is that the pressure distributions and lift coefficient are predicted with much better accuracy than drag coefficient. One of the main contributors to the error in $C_d$ prediction is expected to be due to the contamination by numerical noise in 2-dimensional inviscid simulations. The authors expect that in 3-dimensional simulations, such a problem is less likely to occur and therefore leading to much better predictions for the drag coefficient.

The overall prediction errors of the model are driven by the following main factors (i) POD basis truncation (ii) ROM interpolation errors (iii) DEIM interpolation errors and (iv) finite-precision round-off errors. Among these 4 components, the type of interpolation (in the Euclidean space) plays a very important role with shape parameters. For the NACA test case, the 2nd order multivariate Lagrange interpolation gave best results, while for the RAE test case the 1st order interpolation suited better; suggesting that the best interpolation method for a given problem is highly problem dependent. However, based on the extensive validation process carried out in this work, the 3 outputs ($C_p$, $C_l$, $C_d$) are predicted within 5% given adequate snapshots. This was demonstrated in Renganathan et al (2018). Also (ii) is the only unique source of error compared to other projection-based model reduction methods.

We finally demonstrate the present methodology on applications in many-query context which include global optimization and uncertainty quantification.

3.4 | Inverse Design

In aerodynamic design, a specific aerodynamic load distribution about an aerodynamic object is of interest. For instance, under incompressible flow assumptions for a finite wing, an elliptic lift distribution along the wing is known to produce the least induced drag. Similarly, in the preliminary design of propellers and turbines, a certain lift distribution along the blade is an input to the design process. In such cases the actual design (shape) parameters that produce such a load distribution is of interest. We call such a problem the inverse design. Here we fix a certain pressure coefficient distribution as our target and search the design space for the shape parameters that would best approximate the target. Therefore we are interested in solving the following optimization problem

$$\arg\min_\theta \frac{1}{2} \|C_p(u, \theta) - C_p^*(u, \theta)\|^2_2$$
subject to:
$$R(u, \theta) = 0$$
$$\theta_L \leq \theta \leq \theta_U$$

Where $C_p^*$ is the target pressure distribution. Naturally we want to replace the full-order governing equations with the ROM and hence we solve the modified problem

$$\arg\min_\theta \frac{1}{2} \|C_p(u, \theta) - C_p^*(u, \theta)\|^2_2$$
subject to:
$$\Psi^T R(\Phi \tilde{u}, \theta) = 0$$
$$\theta_L \leq \theta \leq \theta_U$$

The free-stream conditions used for this test case are summarized in Table 3. The target $C_p^*$ and the corresponding airfoil shape is shown in Figure ??, corresponding to a $C_l = 0.5$. The optimum shape was searched using a Genetic Algorithm based optimizer with a population size of 30 per generation and a total of 60 generations. The constraint and function convergence
tolerance were set to $1E-5$ and $1E-3$ respectively and the optimization required a total of 1830 function evaluations of the ROM, to determine the final design. Overall, the optimization required approximately 3.7 hrs of wall-clock time running in serial mode.

The convergence history of GA optimizer is shown in Figure 11; the fitness function ceased to improve significantly beyond 60 generations. There is a discrepancy (in terms of relative error) of about 4.4% between the target and predicted pressure distributions and shapes are shown in Figure 12. Further, it does so within wall-clock time of approximately 3.7 hrs whereas the equivalent FOM wall-clock time for the same number of function evaluations is expected to take roughly 300 hrs. Therefore the computational efficiency achieved via the ROM outweighs the rather small (< 5%) error in its prediction. Having said that, certain amount of error in the prediction of the ROM is always to be expected due to the factors discussed in section 3.3. Overall, this test case demonstrates the usefulness of the ROM for a design optimization setting where it enables fast decision making.

![Optimizer convergence](image1)

**FIGURE 11** Optimizer convergence

### TABLE 3 Free-stream conditions for inverse design

| Parameter | Value |
|-----------|-------|
| $P_\infty$ | 101325 Pa |
| $T_\infty$ | 288 K |
| $\rho_\infty$ | 1.225 kg/m$^3$ |
| $a_\infty$ | 340.296 m/s |
| $M_\infty$ | 0.6 |
| $\alpha$ | 2° |
| $C_l$ | 0.5 |
FIGURE 12 Comparison of predicted-target design (via ROM) with the actual target
3.5 | Uncertainty Quantification

Next, we demonstrate the present methodology towards the uncertainty quantification problem. Uncertainty in aerospace design is inevitable and the decision making process should account for it in order to make more reliable decisions early on in the design process. Specifically, we are interested in quantifying the uncertainty in the airfoil lift and drag coefficients due to the manufacturing process induced variations in the airfoil shape. We use the same ± 30% variation in the airfoil CST coefficients and uniformly sample this design space. A Monte Carlo simulation is carried out to propagate the input uncertainty into the model to quantify the uncertainties in the quantities of interest - namely, the lift and drag coefficients. We use the NACA test case under operating conditions defined in Table 3.

The histograms of the responses are shown in Figures 13 and 14. The ROM histograms are approximated from 4000 samples run via the Monte Carlo simulations. These simulations were run in serial mode on a desktop computer which consumed approximately 7.2 hrs of wall-clock time. An equivalent budget of high-fidelity simulations run in serial for the same level of convergence as the ROM would have cost ≈ 667 hrs of wall-clock time; therefore the computational speed-up with the ROM is ∼ 100x. The FOM histograms were generated from 1000 latin hyper cube samples and represents the truth against which the ROM predictions are compared. Firstly, the histograms of $C_l$ compare very well between the ROM and FOM with a symmetric, normal-like distribution. This is further reflected in the statistics compared in Table 4. In the case of $C_d$ the ROM predicts the right-ward skew of the distribution accurately. The range predicted by the ROM for $C_d$ is approximately [0.005, 0.016] while in the case of the FOM it is a bit narrower, [0.006, 0.01]. The relatively higher sensitivity in the $C_d$ computations compared to $C_l$, as explained in the previous sections, is the main reason for this discrepancy. Further, the number of points with $C_d > 0.01$ fallin in the tail end of the distribution, form small fraction of the total number of samples; which can be considered outliers. Similarly a small fraction of the points fall below 0.006 (lower bound predicted by the FOM) for the same reason. Secondly, the FOM histograms are constructed from a rather coarse(1000) sample of points and hence an exact comparison would require at the very least an equal sample size for both the ROM and FOM. While this might be feasible for the present test case (given its relatively cheaper computational cost when executed with parallel computing) it is avoided because the idea is not scalable for practical problems. We compare them with the only goal of making a high-level observation about the ROM’s capability to capture the actual trends of the FOM.

Overall, the ROM is able to capture general trends such as the range of output quantities of interest and shapes of their distributions with useful accuracy. One needs to be careful in interpreting the statistics compared in Table 4 because both results are an outcome of samples of different sizes while both being a relatively coarse dataset for accurate approximations of distributions. However, they do provide evidence that the ROM has capability to potentially supplement the expensive high-fidelity models while offering compelling computational speed-ups without sacrificing the accuracy significantly.

| Table 4: Comparison of output statistics |
|------------------------------------------|
| $C_l$ | $C_d$ |
| ROM  | FOM | ROM  | FOM |
| Mean | 0.2996 | 0.2980 | 0.0076 | 0.0075 |
| Median | 0.2989 | 0.2969 | 0.0076 | 0.0074 |
| Std. Dev | 0.0587 | 0.0598 | 0.0018 | 0.0007 |
| Skewness | 0.0743 | 0.0431 | 0.2294 | 0.3887 |
| Kurtosis | 2.4794 | 2.7697 | 4.5908 | 2.5210 |
FIGURE 13 Histograms via the ROM. 4000 monte-carlo samples.

FIGURE 14 Histograms via the FOM. 1000 LHC samples
4 | CONCLUSION

A methodology to develop projection-based reduced order models with black-box high-fidelity PDE-based models is developed in this work. We extend the earlier work\textsuperscript{11} to systems with parametric geometry in addition to parametric boundary conditions. The methodology is validated under subsonic and transonic compressible inviscid flow. Under subsonic conditions, exceptional accuracy (<5%) was observed in approximating the flow-field as well as outputs $C_l$, $C_d$ and $C_p$, whereas in the transonic regime the average error is higher (in the 15-20% range). The method suffers from typical limitations of a POD-based methods to capture highly non-linear flow such as ones with moving shocks.\textsuperscript{29,30} A common trend observed in this work is that $C_p$ and $C_l$ are predicted with much better accuracy than $C_d$, which the authors mainly attribute to contamination by numerical noise particularly in 2D inviscid simulations.

Furthermore to demonstrate its computational efficiency, the method is applied to 2 specific applications in the many-query context: (i) inverse design and (ii) uncertainty quantification via Monte Carlo analysis. In both situations, the ROM is executed $\mathcal{O}(1000)$ times at a wall-clock time of 2-8 hrs, while the equivalent budget of FOM would have consumed 200-800 hrs. Therefore first and foremost, these results demonstrate the utility of the methodology towards real-time decision making. The inverse design test case is mainly the test of the ROM to satisfy the physics of the problem, even in its approximated form. The predictions of the ROM is within 5% of the actual solution while again, achieving it at superior computational efficiency compared to the FOM. Lastly, a Monte Carlo analysis with 4000 uniformly sampled points from the input space was used to approximate the probability distributions of the two main outputs considered in this work: the $C_d$ and $C_l$. Overall, the $C_l$ showed better match with the FOM results, while the $C_d$ predicted the statistics with relatively higher discrepancy. Regardless, the predictions of the ROM turn out to capture general trends at a fraction of the computational cost of the FOM.

The present work is a first step towards performing projection-based model reduction with black-box models. Therefore to keep the exposition simple, the 2-dimensional compressible euler equations are used as the test case. However, the methodology naturally extends to 3-dimensional systems with $S > 4$ without modification. The only requirement is complete knowledge of the governing equations in continuous PDE form, which is always available in the form of code documentation with black-box commercial codes. The authors are currently investigating purely data-driven methods where this requirement can be waived. Another direction for future work is the adaptive construction of the model, where snapshots are sequentially generated based on certain greedy criterion.

Overall, the present methodology establishes feasibility for projection based MOR for static parametric systems available as a black-box. Since at the time of this writing, no other competing method exists to cater exactly the same category of systems, comparison against a competing method is not performed in this work.
References

1. Johnson Forrester T., Tinoco Edward N., Yu N. Jong. Thirty years of development and application of CFD at Boeing Commercial Airplanes, Seattle. *Computers and Fluids.* 2005;34(10):1115–1151.

2. Holmes, Philip., Lumley, John L., Berkooz, Gahl and Rowley Clarence W. *Turbulence, Coherent Structures, Dynamical Systems and Symmetry.* 1998.

3. Xiao D, Fang F, Buchan A G, Pain C C, Navon I M, Muggeridge A. Non-intrusive reduced order modelling of the Navier Stokes equations. *Computer Methods in Applied Mechanics and Engineering.* 2015;293:522–541.

4. Christophe Audouze Florian De Vuyst, Nair Prasanth B.. Nonin intrusive Reduced-Order Modeling of Parametrized Time-Dependent Partial Differential Equations. *Numerical Methods for Partial Differential Equation.* 2013;29(5):1587–1628.

5. Audouze C. De Vuyst F., B. Nair P.. Reduced-order modeling of parameterized PDEs using time-space parameter principal component analysis. *INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN ENGINEERING.* 2009;80:1025–1057.

6. Bui-Thanh Tan, Damodaran Murali, Willcox Karen E.. Aerodynamic Data Reconstruction and Inverse Design Using Proper Orthogonal Decomposition. *AIAA Journal.* 2004;42(8):1505–1516.

7. Brunton Steven L, Proctor Joshua L, Kutz J Nathan. Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *Proceedings of the National Academy of Sciences.* 2016;201517384.

8. Peherstorfer Benjamin, Willcox Karen. Data-driven operator inference for nonintrusive projection-based model reduction. *Computer Methods in Applied Mechanics and Engineering.* 2016;306:196–215.

9. Koopman Bernard O. Hamiltonian systems and transformation in Hilbert space. *Proceedings of the National Academy of Sciences.* 1931;17(5):315–318.

10. Renganathan S. Ashwin.. A Methodology for Non-Intrusive projection-based model reduction of expensive black-box PDE-based systems and application in the many-query context. PhD thesisGeorgia Institute of Technology2018.

11. Renganathan S. Ashwin Yingjie, Mavris Dimitri N.. Koopman-Based Approach to Nonintrusive Projection-Based Reduced-Order Modeling with Black-Box High-Fidelity Models. *AIAA Journal (d.o.i. 10.2514/1.J056812).* 2018;1–25.

12. Williams Matthew O., Kevrekidis Ioannis G., Rowley Clarence W.. A Data-Driven Approximation of the Koopman Operator: Extending Dynamic Mode Decomposition. *Journal of Nonlinear Science.* 2015;25(6):1307–1346.

13. Tu Jonathan H, Rowley Clarence W, Luchtenburg Dirk M, Brunton Steven L, Kutz J Nathan. On dynamic mode decomposition - theory and applications. *Journal of computational dynamics.* 2013;(September):1–30.

14. SCHMID PETER J.. Dynamic mode decomposition of numerical and experimental data. *Journal of Fluid Mechanics.* 2010;656:5–28.

15. Kutz J Nathan, Proctor Joshua L., Brunton Steven L.. Generalizing Koopman theory to allow for inputs and control. *arXiv:1602.07647v1.* 2016;1–21.

16. Kutz Nathan, Proctor Joshua L, Brunton Steven L. Koopman Theory for Partial Differential Equations. *arXiv:1607.07076v1.* 2016;1–21.

17. Brunton Steven L., Brunton Bingni W., Proctor Joshua L., Kutz J. Nathan. Koopman invariant subspaces and finite linear representations of nonlinear dynamical systems for control. *PLoS ONE.* 2016;11(2):1–19.

18. Chaturantabut Saifon, Sorensen Danny C.. Nonlinear Model Reduction via Discrete Empirical Interpolation. *SIAM Journal on Scientific Computing.* 2010;32(5):2737–2764.

19. Golub Gene H., Van Loan Charles F.. *Matrix computations.* JHU Press; 2012.
20. Rahman Inam Ur, Drori Iddo, Stodden Victoria C., Donoho David L., Schröder Peter. Multiscale Representations for Manifold-Valued Data. *Multiscale Modeling & Simulation*. 2005;4(4):1201–1232.

21. Barachant Alexandre, Bonnet Stéphane, Congedo Marco, Jutten Christian. Riemannian geometry applied to BCI classification. In: . Springer; 2010.

22. STARCCM+ url: http://mdx.plm.automation.siemens.com/star-ccm-plus. 2017.

23. Kulfan Brenda, Bussoletti John. Fundamental Parametric Geometry Representations for Aircraft Component Shapes. In: . 2006.

24. Kulfan Brenda M. Universal parametric geometry representation method. *Journal of Aircraft*. 2008;45(1):142–158.

25. Anderson Jr John David. *Fundamentals of aerodynamics*. Tata McGraw-Hill Education; 2010.

26. Adkins Charles N, Liebeck Robert H. Design of optimum propellers. *Journal of Propulsion and Power*. 1994;10(5):676–682.

27. Renganathan S. Ashwin, Denney Russell K, Duquerrois Antoine, Mavris Dimitri N. Validation and Assesment of Lower Order Aerodynamics Based Design of Ram Air Turbines. In: . 2014.

28. Mooney Christopher Z. *Monte carlo simulation*. Sage Publications; 1997.

29. Beran Philip S, Silva Walter. Reduced Order Modeling: New Approaches for Computational Physics. *42rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*. 2001;.

30. LeGresley Patrick Allen. Application of proper orthogonal decomposition (POD) to design decomposition methods. PhD thesis2006.

31. Tu Jonathan H, Rowley Clarence W, Luchtenburg Dirk M, Brunton Steven L, Kutz J Nathan. On Dynamic Mode Decomposition: Theory and Applications. *(Dmd)*:1–30.

32. Schittkowski Klaus. NLPQL: A FORTRAN subroutine solving constrained nonlinear programming problems. *Annals of operations research*. 1986;5(2):485–500.

33. Poirier Diane, Allmaras Steven R, McCarthy Douglas R, Smith Matthew F, Enomoto Francis Y. The CGNS system. *AIAA paper*. 1998;(98-3007).

### 5 | APPENDIX

#### 5.1 | Shape parameterization

The CST model of parameterization defines a *class* function $C$ and a *shape* function $S$ and the curve being parameterized is specified as their product. The class function, $C(\psi)$ is more generally defined as

$$C_n^m(\psi) := \psi^n(1 - \psi)^m$$

where the variable $\psi$ represents the non-dimensional chord-wise distance, $n_1$ and $n_2$ define the specific class. The unique shape of an airfoil is driven by the shape function, specified as follows

$$S(\psi) = \sum_{i=0}^{n} A_i \psi^i$$

(26)

It is particularly useful to define a *unit shape function*, i.e. $S(\psi) = 1$ such that the individual coefficients $A_i$ can be obtained as generic constants. For instance for $n = 1$ the simplest decomposition one could get for the shape function is $S(\psi) = S_0(\psi) +$ 

---

$A_i$'s are denoted by $K_{i,n}$ for unit shape functions.
where

\[ S_1(\psi) = \psi \quad \text{and} \quad S_2(\psi) = 1 - \psi \]

where the coefficients \( A_0 = 1 \) and \( A_1 = 1 \). Similarly, for the general \( n \)th order shape function, the decomposition of the unit shape function can be done using Bernstein polynomials

\[ S(\psi) = \sum_{i=0}^{n} K_{i,n} \psi^i (1 - \psi)^{n-i} \quad (27) \]

where the coefficients are the binomial coefficients given by

\[ K_{i,n} = \binom{n}{i} = \frac{n!}{i!(n-i)!} \]

The final shape of the airfoil shape is then given by

\[ y(\psi) = C(\psi) S(\psi) \quad (28) \]

The unit shape functions and the corresponding airfoil geometries are illustrated in the Figure 15. It can be seen that such a parametrization results in each component shape function peak being equally distributed between the leading and trailing edges leading to the same effect in the component airfoils. It is now a matter of scaling up or down, the binomial coefficients of the Bernstein polynomials in order to approximate the unique airfoil shape of interest.

The coefficients \( A_i \) represent the actual parameters of the shape, given \( n \) the order of the Bernstein polynomials. An \( n \)th order CST parametrization has \( n + 1 \) parameters. If separate parametrizations are sought for the upper and lower surfaces of the airfoil, then the CST parametrization leads to \( 2(n + 1) \) parameters to specify the whole shape of the airfoil, where the \( n \) needs to be determined for a specific geometry under consideration. However, typically \( n = 3 \) – 5 are observed to be adequate to parametrize the airfoil shapes considered in this work. One way to determine \( n \) and the associated polynomial coefficients is

**FIGURE 15** Examples of the shape function decomposition (into Bernstein polynomials) for various values of the order \( n \) (left) and the resulting component airfoils (right). The coefficients \( A_i \) correspond to unit shape function, which can be scaled up/down to obtain a specific airfoil shape.
to find the values that minimizes certain error between the true shape of the airfoil and the resulting approximation via CST. In
this work, the parameters for a given airfoil shape are determined by solving the following minimization problem

$$\min_{A_j, n} \left\| y(\bar{\psi}) - C(\bar{\psi}) S(\bar{\psi}, A) \right\|^2$$

(29)

where $\bar{\psi} \in \mathbb{R}^{n+1}$ are $n + 1$ equally spaced points sampled from $\psi$ spanning $[0, 1]^2$. This way, the smallest possible $n$ and their corresponding Bernstein coefficients are determined. For the RAE2822 airfoil shape, the following parameterization was obtained ($n = 3$):

$$A_{RAE2822} = \begin{bmatrix} 0.1268 & 0.4670 & 0.5834 & 0.2103 \\ -0.1268 & -0.5425 & -0.5096 & 0.0581 \end{bmatrix}$$

where the first and second rows represent the parameterization of the upper & lower surfaces of the airfoil; the comparison of the CST curve and the actual RAE2822 shape is shown in Figure ?? It can be seen that the CST parametrization gives an adequate approximation to the true curve with only 8 parameters. The coefficients may now be perturbed to modify the baseline airfoil shape.

Similarly, the NACA0012 airfoil shape is approximated via CST and is also shown in Figure ?? In this case, due to the lack of camber, the CST gives very good approximation with $n = 2$. Additionally, due to the symmetry of the airfoil about the chord, the parameters (given below) are equal in magnitude and opposite in sign. However, all the $2(n + 1)$ degrees of freedom are considered in this work for the NACA0012 airfoil.

$$A_{NACA0012} = \begin{bmatrix} 0.1689 & 0.2699 & 0.1387 \\ -0.1689 & -0.2699 & -0.1387 \end{bmatrix}$$

Note that picking equally spaced points works well for the current airfoil geometries under consideration in this work and may not work for any arbitrary geometry. More generically, a least-squares fit considering all the points might be more suitable.
### TABLE 5 Comparison of $C_p$, $C_l$ and $C_d$ between ROM & FOM for the NACA0012 test case

| Case | $C_p$ Error % | $C_l$ (ROM) | $C_l$ (FOM) | Error % |
|------|---------------|-------------|-------------|---------|
| 1    | 1.29          | 0.1889      | 0.1912      | 1.20    |
| 2    | 0.74          | 0.2018      | 0.2070      | 2.50    |
| 3    | 0.80          | 0.2932      | 0.2943      | 0.37    |
| 4    | 1.86          | 0.2795      | 0.2865      | 2.44    |
| 5    | 1.36          | 0.3550      | 0.3621      | 1.96    |
| 6    | 0.62          | 0.3691      | 0.3664      | 0.73    |
| 7    | 0.46          | 0.3298      | 0.3272      | 0.79    |
| 8    | 2.79          | 0.3229      | 0.3312      | 2.50    |
| 9    | 0.76          | 0.3109      | 0.3137      | 0.89    |
| 10   | 5.43          | 0.2710      | 0.3065      | 11.58   |

### TABLE 6 Comparison of $C_p$, $C_l$ and $C_d$ between ROM & FOM for the RAE2822 test case

| Case | $C_p$ Error % | $C_d$ (ROM) | $C_d$ (FOM) | Error % | $C_l$ (ROM) | $C_l$ (FOM) | Error % |
|------|---------------|-------------|-------------|---------|-------------|-------------|---------|
| 1    | 7.68          | 0.0161      | 0.0174      | 7.47    | 0.9174      | 0.9608      | 4.52    |
| 2    | 12.93         | 0.0336      | 0.0302      | 11.26   | 1.0669      | 0.9825      | 8.59    |
| 3    | 2.84          | 0.0257      | 0.0262      | 1.91    | 1.1140      | 1.1446      | 2.67    |
| 4    | 8.14          | 0.0321      | 0.0264      | 21.59   | 1.1004      | 1.0901      | 0.95    |
| 5    | 13.77         | 0.0194      | 0.0224      | 13.39   | 0.8965      | 1.0345      | 13.34   |
| 6    | 12.78         | 0.0298      | 0.0484      | 38.43   | 1.0288      | 1.0082      | 2.04    |
| 7    | 5.91          | 0.0306      | 0.0279      | 9.68    | 0.9403      | 0.9286      | 1.26    |
| 8    | 4.99          | 0.0245      | 0.0245      | 0       | 0.9124      | 0.8914      | 2.36    |
| 9    | 5.63          | 0.0292      | 0.0315      | 7.30    | 0.9507      | 0.9779      | 2.78    |
| 10   | 9.96          | 0.0326      | 0.0217      | 50.23   | 0.9750      | 0.9793      | 0.44    |

### 5.2 Model validation data

The values of the output quantities of interest and their associated errors (defined by Eq. 20) is presented in Tables 5 and 6 below.
5.3 | Discrete Empirical Interpolation Method (DEIM)

The Discrete Empirical Interpolation Method (DEIM) is briefly reviewed here and as an illustration one of the non-linear constraints used in Eq. [19] is evaluated. For a non-linear function \( f(\theta) \in \mathbb{R}^N \) the DEIM approximates \( f \) by projecting it onto a subspace spanned by \( \{x_1, \ldots, x_q\} \subset \mathbb{R}^N \) as

\[
f(\theta) \approx Xc(\theta)
\] (30)

where \( X = [x_1, \ldots, x_q] \in \mathbb{R}^{N \times q}, \ q \ll N \) is determined via a POD of the snapshots of \( f \) and is assumed to be globally valid in the design space that bounds the design parameters \( \theta \) and \( c(\theta) \in \mathbb{R}^q \) are the coefficients of the basis expansion. Then the approximation of \( f \) requires only the determination of \( c(\theta) \) which requires only \( q \) equations. The DEIM gives a distinguished set of \( q \) points from the over-determined system \( f(\theta) = Xc(\theta) \). Given a permutation matrix \( P \) that would give \( q \) such distinguished rows of a matrix when pre-multiplied, then the \( q \times q \) system necessary to solve for the coefficients is given by

\[
P^Tf(\theta) = (P^TX)c(\theta)
\] (31)

So the approximation of \( f(\theta) \) is then given by

\[
f(\theta) \approx X(P^TX)^{-1}P^Tf(\theta)
\] (32)

If the \( q \) row-indices (that are extracted by pre-multiplying with \( P^T \)) are represented by a vector, \( q \), then in the above equation, \( P^Tf(\theta) \) is equivalent to extracting the \( q \) rows of \( f \). Therefore the approximation of \( f(\theta) \) requires only \( q \) computations which is efficient because \( q \ll N \). Similarly, a non-linear function that depends on the state, \( f(u) \) can be approximated as

\[
f(u) \approx X(P^TX)^{-1}P^Tf(u)
\] (33)

Since \( u = \Phi_k^T\tilde{u} \) and setting \( \tilde{f} = \Phi_k^Tf(u) \), \( \tilde{f} \) can be approximated as

\[
\tilde{f} = \Phi_k^TX(P^TX)^{-1}f(P^T\Phi_k\tilde{u})
\] (34)

In the above equation, the term \( \Phi_k^TX(P^TX)^{-1} \) is independent of the state and hence can be pre-computed and \( P^T\Phi_k \) is just extraction of the \( q \) rows of \( \Phi_k \). Therefore using the DEIM, the non-linear term can be expressed in terms of the reduced state, \( \tilde{u} \) and hence can be efficiently computed.

Now the DEIM is illustrated on evaluating the first constraint of Equation [19] which in discretized form is given below

\[
h_1 = y_3 - \frac{y_1y_3}{y_2}
\] (35)

Let \( \rho \) be the vector containing the \( q \) row-indices returned by DEIM via snapshots of the non-linear term \( y_3 \) and \( \Phi_1, \Phi_2, \Phi_3 \) be the projection matrix of \( y_1, y_2, y_3 \) and \( y_5 \) respectively. Then

\[
\hat{h}_1 = y_5 - Ph_i^TX [X(\rho_3, :)^{-1}]^{-1} \left\{ \frac{\Phi_1(\rho_3, :)y_1}{\Phi_2(\rho_3, :)y_2} \right\}
\] (36)

In the above equation, the term outside of the braces can be pre-computed. Additionally since \( y_5 = \frac{x_5}{y_2} \), \( X = \Phi_5 \) and hence the term reduces to \( [X(\rho_3, :)^{-1}]^{-1} \) which is \( q \times q \) and hence can be cheaply computed. Therefore using the DEIM, the non-linear constraints are evaluated in terms of the reduced state variables which makes it computationally cheap.

**AUTHOR BIOGRAPHY**