Nonlinear Reduced Order Modeling using Domain Decomposition

Nikhil Iyengar[*], Dushhyanth Rajaram[†], Kenneth Decker[‡], Christian Perron[§], and Dimitri Mavris[¶]
Aerospace Systems Design Laboratory, Georgia Institute of Technology, Atlanta, GA, 30332 USA

As designers become increasingly reliant upon expensive, high-fidelity numerical modeling and simulation, Reduced Order Modeling (ROM) has emerged as a compelling method of predicting high-dimensional discretized field outputs in a compact and efficient manner. This study presents a parametric, non-intrusive ROM which uses Domain Decomposition (DD) and is capable of employing multiple dimension reduction techniques within different spatial sub-domains of the field. To enforce smoothness of solutions across sub-domains, points at domain interfaces are masked and the Gappy Proper Orthogonal Decomposition (Gappy-POD) method is used to reconstruct the global solution. The methodology is assessed on three test cases, including two-dimensional turbulent flow around the RAE2822. Results show that nonlinear ROMs predict fields more accurately in the vicinity of discontinuous features compared to their linear counterparts, which are more accurate in regions that are far from these discontinuities. Furthermore, application of nonlinear ROMs only in the vicinity of a discontinuity using DD resulted in predictive error reduction near the shockwave and significant improvement in total error compared to existing approaches that employ a single global model. Finally, Gappy-POD was shown to provide smooth field predictions with linear DDROMs, though its performance deteriorated with nonlinear methods.

NOMENCLATURE

C_p	=	Pressure coefficient
$\mathbf{a}$	=	Gappy-POD expansion coefficient vector
b	=	Number of design variables
d	=	ROM dimension
d_{ij}	=	Pairwise distance operator
e	=	Relative error
g	=	Generic functions
i, j, k, l	=	Generic indices
k	=	Number of nearest neighbors
M	=	Mach number
$\mathbf{m}$	=	Gappy-POD mask vector
n	=	Dimension of the high-fidelity fields
$\mathbf{p}$	=	Vector of design parameters
q	=	Generic sub-region index
S	=	Number of sub-domains
v	=	Number of validation snapshots
w	=	Weight coefficients
$\mathbf{y}$	=	High-fidelity snapshot with missing information
$\mathbf{Z}$	=	Matrix of high-fidelity points projected on latent space
$\mathbf{z}$	=	Coordinates of single high-fidelity point in latent space
$\mathbf{X}$	=	Matrix of high-fidelity snapshots
$\mathbf{x}$	=	Single high-fidelity snapshot

[*] Ph.D. Student, School of Aerospace Engineering (ASDL), Georgia Tech, Student AIAA Member
[†] Research Engineer, School of Aerospace Engineering (ASDL), Georgia Tech, AIAA Member
[‡] Research Engineer, School of Aerospace Engineering (ASDL), Georgia Tech, Student AIAA Member
[§] Postdoctoral Researcher, School of Aerospace Engineering (ASDL), Georgia Tech, AIAA Member
[¶] S.P. Langley Distinguished Regents Professor and Director of ASDL, Georgia Tech, AIAA Fellow

α	=	Angle of attack
β	=	Shock angle
γ	=	Model tuning parameter
ϵ	=	Cost function weight
θ	=	Generic angle
σ	=	Singular value
ϕ	=	POD mode
Φ	=	Matrix of POD modes
Ω	=	Computational domain

ANN	=	Artificial Neural Network
CFD	=	Computational Fluid Dynamics
DD	=	Domain Decomposition
DDROM	=	Domain Decomposition-based Reduced Order Modeling
DoE	=	Design of Experiments
DR	=	Dimensionality Reduction
DV	=	Design Variable
FFD	=	Free-Form Deformation
FOM	=	Full Order Model
ISOMAP	=	Isometric Feature Mapping
LLE	=	Locally Linear Embedding
NLDR	=	Non-Linear Dimensionality Reduction
POD	=	Proper Orthogonal Decomposition
RANS	=	Reynolds Average Navier-Stokes
RBF	=	Radial Basis Function
RIC	=	Relative Information Content
ROM	=	Reduced Order Model
SVD	=	Singular Value Decomposition
TPS	=	Thin-Plate Spline

I. Introduction

The design of complex, engineered systems from conceptualization to commercialization is expensive, with aircraft design programs costing billions of dollars and often spanning more than a decade. Historically, knowledge about the design and cost is accrued progressively during the prototype development and testing phase rendering subsequent changes to the design costly and time-inefficient. Today, designers have access to physics-based modeling and simulation tools that can provide realistic estimations of system behavior and performance at a small fraction of the time and cost of physical experimentation.

However, the design and development of novel aircraft concepts often require on the order of millions of expensive simulations [1]. Inexpensive, low-fidelity tools are a popular alternative, trading accuracy for computational efficiency; but during the design of unconventional aircraft, such as low-boom supersonic transports, low fidelity tools may not capture the complex shock interactions and nonlinear flow features needed to properly assess system performance [2, 3]. While there has been advancements in computational power, the computational cost and labor necessary to implement high-fidelity numerical simulations capable of capturing the complex physics has imposed practical constraints on their use. To address this challenge, recent studies have identified full field surrogate models or Reduced Order Models (ROMs) as a key enabler for leveraging high-fidelity numerical simulations earlier in the design process [4].

A. Reduced-Order Modeling

With data volumes rapidly growing, there has been a significant push to develop the mathematical tools necessary to distill physics from data and rapidly perform simulations [5–7]. As seen in a number of recent studies, ROMs have served as promising candidates for bringing high-fidelity numerical simulation data earlier into the design process in a compact and computationally efficient manner [8–11]. The central assumption behind ROM is that there exists a low-dimensional subspace or *latent space* that is capable of representing the fundamental features of the high-dimensional data obtained from a physical problem. ROM attempts to identify such a low dimensional subspace, project the original high-dimensional data into that subspace, and predict solutions at unseen data points for a fraction of the original cost.

A commonly used dimension reduction (DR) method for ROM is the Proper Orthogonal Decomposition (POD). Introduced by Lumley in the 1960s [12], POD has been successfully used in various aerodynamic applications [4, 13–17]. Using a set of snapshots obtained from a high fidelity simulation tool, like CFD, the POD process computes a set of eigenfunctions that minimizes the error between the original data and reconstructed data [18]. However, flows with highly nonlinear features (such as shocks) have been demonstrated to challenge the assumption of linearity that is inherent in POD and result in models with high error, particularly in the vicinity of such features [9, 19, 20]. While this error is not significant in the design of low speed aircraft, the design of supersonic and high speed transonic aircraft is extremely dependent on shocks and require accurate modeling of these features [21].

To overcome this deficiency of POD-based ROMs, recent studies have explored the use of transforming data before the application of POD and nonlinear DR, such as manifold learning, as alternatives. In [22], the solution is decomposed into an evolution term to capture moving shocks and a diffusion term to capture the changing shapes. Nair et al. [23] developed the transported snapshot model order reduction method which projects the governing equations on a set of snapshots transformed spatially. Other studies also pursued a similar snapshot transformation method [24, 25]. But neither extended this approach to parameter dependent shocks and both were restricted to models with a few input parameters. Nonlinear DR methods have recently been shown to potentially capture discontinuous flow features, such as shocks, more effectively than their linear counterparts and handle many input parameters [20, 26]. However, Decker et al. [19] showed that while nonlinear ROMs generally outperformed linear ROMs near the discontinuity, the latter were often observed to have higher accuracy in regions of the field that were far from discontinuities.

To improve the accuracy of linear ROMs near shocks, a Domain Decomposition (DD) strategy has been used in previous studies. In this technique, the fluid domain is spatially separated into regions where the shock (or discontinuity) is isolated. Then, either a hybrid full-order/reduced-order or complete reduced order method is used. When introduced by Lucia et al. [27, 28], the DD strategy resorted to solving the actual full-order model in the region of the shock, while POD was used everywhere else. Later approaches relied mostly on intrusive methods, which required access to the governing equations or the full-order model's code, to perform DD-based ROM (DDROM) [29]. Recently in [30], the authors used a hyper-reduction strategy where the Petrov-Galerkin projection was used to obtain the reduced order equations. While many of the above methods showed better shock prediction than using POD over the entire solution domain, most often, the code and implementation used to solve the CFD problem is proprietary and unavailable to the user. Recent work [31, 32] has relied on a completely non-intrusive approach to DDROM. While this was an essential step toward a purely data-driven approach, both of these studies relied solely on POD in each sub-domain, which was still unable to accurately capture the shock discontinuity.

B. Challenges with Domain Decomposition

One common challenge associated with DDROM is large gradients appear at the interfaces between sub-regions. To tackle this issue, most DDROM studies in the past have resorted to an intrusive approach to ensure smoothness of solution from one region to the next. In [27], a constrained optimization problem is solved to ensure that the error at overlapping fluid cells is minimized. In [30], equality of velocity is enforced at overlapping regions between two sub-domains by penalizing nodes where velocities do not match. In [33], where DD is applied to solid mechanics problems, interface continuity is ensured by enforcing equality of displacements at the interface of two sub-domains. These last two examples both rely on an intrusive, Petrov-Galerkin projection approach.

This work focuses on models that are entirely non-intrusive because of their robustness and practicality for use with black-box analysis tools. The issue of ensuring smoothness at boundary interfaces is rich with literature based on intrusive solutions. However, the use of non-intrusive methods remains relatively unexplored. Indeed, only a few recent studies have explored non-intrusive techniques to ensure smoothness between sub-domains: Xiao et al. [32] used a multi-dimensional radial basis function interpolation method to construct a set of hypersurfaces for a given domain. This approach was exclusively restricted to unsteady simulations and required the solution at each time step to enforce boundary conditions between domains. Sun et al. [31] used Hermite polynomials to interpolate the solution between two regions.

One reason ROMs show superior performance to data-fit interpolation models is because ROMs retain the underlying mathematical structure of the system [34]. It should follow that a method that relies on the high-fidelity field output data to capture the underlying physics and ensure smoothness of derivatives across sub-domain interfaces should improve the global predictions of DDROM. In the present study, the predicted solution between two neighboring regions is computed using the Gappy-POD approach [35–37]. This work hypothesizes that the reconstructed solution at the interfaces will have reduced gradients because Gappy-POD uses the global POD basis (which accounts for the underlying physics [38]) and a least-squares optimization procedure to minimize the change in values between sub-domains. However, as Gappy-POD uses a linear Dimensionality Reduction (DR) method for reconstruction, there will be a mismatch with the local DDROM bases. The impact of this mismatch, particularly on nonlinear DDROMs, will be empirically investigated in this work.

C. Contributions of the Paper and Structure

The goal of this work is to investigate and quantify the ability of DDROM methods to accurately and efficiently predict aerodynamic flow fields, with an emphasis on field outputs containing discontinuous features such as shocks. This paper presents a parametric, non-intrusive, reduced-order modeling method that leverages both linear and non-linear DR. In particular, by using manifold learning (a type of nonlinear DR method) in regions where discontinuities are present and POD everywhere else, it is hypothesized that both the global predictive error and error near discontinuities can be decreased relative to that of global ROMs. This work also characterizes the impact of DD on ROMs developed using only nonlinear or linear DR. As previous literature has been restricted to POD-based DDROMs, this will provide empirical insight into domain decomposition's behavior when used with nonlinear DR methods. Finally, the paper applies Gappy-POD for smoothening the solutions at sub-domain interfaces and discusses its qualitative and quantitative impact on the predicted solution. The behavior of the proposed ROM method is evaluated by performing a thorough empirical assessment on datasets from a number of test cases, including a transonic CFD application problem.

The structure of the manuscript is as follows: Section II presents an overview of POD, Gappy-POD, and manifold learning. Section III provides details of the proposed DDROM method. Section IV describes the numerical experiments and results for three test cases: 2D supersonic flow past a wedge, quasi-1D flow through a converging-diverging nozzle, and 2D transonic flow over an airfoil. Section V summarizes the findings and recommendations for future work.

II. Reduced Order Modeling

This section provides the mathematical formulations for the methods implemented in this study. An overview of POD is provided first, followed by a summary of the Gappy-POD and manifold learning algorithms.

A. Proper Orthogonal Decomposition

Due to its simplicity and versatility, for decades, the POD method has been used commonly not only in aerospace applications, but also in many other fields, such as computer science [39]. Though it is often referred to as POD in ROM literature, it is also known as Principal Component Analysis (PCA) or Karhunen-Loève Decomposition.

Consider a matrix $\mathbf{X}$ that is composed of m samples, or *snapshots*, of a field of dimension n obtained from a numerical simulation such that $\mathbf{X} = \left[\mathbf{x}^1, ..., \mathbf{x}^m\right] \in \mathbb{R}^{n\times m}$. For convenience and without loss of generality, assume the matrix $\mathbf{X}$ has been mean-centered as a pre-processing step such that

$$\bar{\mathbf{x}} = \frac{1}{m}\sum_{j}^{m}\mathbf{x}^j = 0 \tag{1}$$

The POD algorithm seeks to project the discretized, high-dimensional data in $\mathbf{X}$ into a low-dimensional linear subspace spanned by a set of orthonormal basis vectors, or *modes*. Let the matrix $\mathbf{\Phi}$ be defined such that its columns consist of this set of orthonormal modes, $\mathbf{\Phi} = \left[\boldsymbol{\phi}^1, ..., \boldsymbol{\phi}^d\right] \in \mathbb{R}^{n\times d}$, where d is the rank of the orthonormal basis and thus the dimension of the latent space. The matrix $\mathbf{\Phi}$ is obtained by performing a Singular Value Decomposition (SVD) of the matrix $\mathbf{X}$

$$\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T \tag{2}$$

where $\mathbf{U} \in \mathbb{R}^{n\times n}$, $\mathbf{V} \in \mathbb{R}^{m\times m}$, and $\mathbf{\Sigma} = diag(\sigma_1, ..., \sigma_r) \in \mathbb{R}^{n\times m}$ such that $\sigma_1 \geq \sigma_2 \geq ... \geq \sigma_r \geq 0$ and $r = min(n, m)$. The eigenvectors of $\frac{1}{m}\mathbf{X}^T\mathbf{X}$ and $\frac{1}{m}\mathbf{X}\mathbf{X}^T$ are given by the columns of $\mathbf{V}$ and $\mathbf{U}$ respectively and the corresponding eigenvalues are given by the values $\sigma_1^2, ..., \sigma_r^2$. From this property, the modes $\boldsymbol{\phi}^j$ can be obtained from the columns of $\mathbf{U}$. A lower rank approximation of the original data can be obtained by computing a matrix $\mathbf{U}' \in \mathbb{R}^{n\times d}$ with a rank of d such that $d < r$ [40]. The reduced rank matrix, $\mathbf{U}'$ is then obtained by preserving the first d columns of $\mathbf{U}$, which correspond to the d largest singular values in $\mathbf{\Sigma}$. The modes obtained from POD are uncorrelated (orthogonal) and maximize the variance captured from the original data [41]. It is possible to define the Relative Information Content (RIC) of including d basis vectors in a truncated model as follows [40]:

$$RIC(d) = \frac{\sum_{i=1}^{d} \sigma_i^2}{\sum_{j=1}^{r} \sigma_j^2} \tag{3}$$

Typically, the number of basis vectors d preserved in a POD approximation is not known a priori. Instead, the practitioner first selects a desired information content $\delta \in [0, 1]$ to preserve, then selects d basis vectors such that $RIC(d) \geq \delta$, with values of $\delta \geq .9999$ seen commonly in application [42]. Once the modes are determined, the coordinates of the training data in the latent space can be obtained by the projection $\mathbf{Z} = \mathbf{\Phi}^T \mathbf{X}$, where $\mathbf{Z}$ is the matrix whose columns correspond to each snapshot's coordinates in the latent space, such that $\mathbf{Z} = [\mathbf{z}^1, ..., \mathbf{z}^m] \in \mathbb{R}^{d\times m}$.

B. Gappy-POD

In the Gappy-POD procedure, a global POD basis is obtained and subsequently used to find the optimal reconstruction of a vector with missing entries. Originally introduced by Everson et al. [35], the procedure is as follows: Let $\mathbf{y} \in \mathbb{R}^n$ be some solution at an unknown point with some missing information. Define a mask vector, $\mathbf{m}$, such that

$$\mathbf{m}_i = \begin{cases} 1, & \text{if } \mathbf{y}_i \text{ is known} \\ 0, & \text{if } \mathbf{y}_i \text{ is missing} \end{cases} \tag{4}$$

where $\mathbf{m}_i$ is the ith element of vector $\mathbf{m}$. Using the POD modes, $\mathbf{\Phi}$, the challenge is to determine a set of coefficients $\tilde{a}_k$ such that a repaired solution (or solution with imputed missing entries), $\tilde{\mathbf{y}}$, can be constructed as follows:

$$\tilde{\mathbf{y}} = \sum_{j=1}^{k} \tilde{a}_j \boldsymbol{\phi}^j \tag{5}$$

Since $\mathbf{y}$ has only a subset of components which are known, a *gappy inner product* is defined as

$$(\mathbf{u}, \mathbf{v})_m := \sum_{i=1}^{n} \mathbf{m}_i \mathbf{u}_i \mathbf{v}_i \tag{6}$$

Using the gappy norm, $\tilde{a}_k$ can be determined using a least-squares algorithm to minimize the error

$$E = (\mathbf{y} - \tilde{\mathbf{y}}, \mathbf{y} - \tilde{\mathbf{y}})_m \tag{7}$$

The coefficients can then be found by differentiating Equation 7 with respect to each coefficient, $\tilde{a}_k$. This leads to the linear system of equations

$$\mathbf{M}\tilde{\mathbf{a}} = \mathbf{f} \tag{8}$$

where

$$M_{ij} = (\boldsymbol{\phi}^{\boldsymbol{i}}, \boldsymbol{\phi}^{\boldsymbol{j}})_m \tag{9}$$

and

$$\mathbf{f}_i = (\mathbf{y}, \boldsymbol{\phi}^{\boldsymbol{i}})_m \tag{10}$$

If the number of known elements in $\mathbf{y}$ is sufficiently large (measurement space is sufficiently dense), then $\mathbf{M}$ approaches $\mathbf{I}$ and $\tilde{\mathbf{a}}$ converges to $\mathbf{a}$, the true coefficient vector for the POD expansion. Once the coefficients, $\tilde{a}_k$, are determined, the full solution can be reconstructed by

$$\mathbf{y}_\mathbf{i}^* = \begin{cases} \mathbf{\Phi}\tilde{\mathbf{a}}, & \mathbf{m}_i = 0 \\ \mathbf{y}_\mathbf{i}, & \mathbf{m}_i = 1 \end{cases} \tag{11}$$

C. Manifold Learning

Manifold learning algorithms are a family of Nonlinear Dimensionality Reduction (NLDR) methods that are used to identify a low-dimensional nonlinear manifold that accounts for the observed variability in a set of high-dimensional data [43]. The fundamental goal of manifold learning algorithms is to solve the embedding problem, which is stated as follows by Bernstein and Kuleshov:

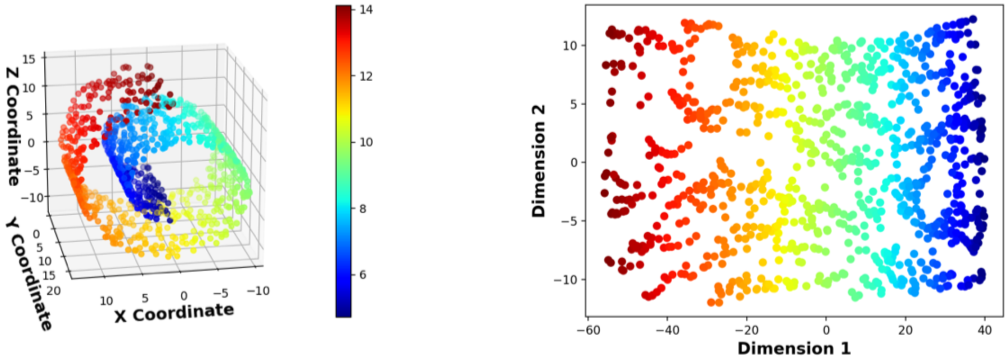
> For a given data set $\mathbf{X} = \left[\mathbf{x}^1, ..., \mathbf{x}^m\right] \subset \mathcal{X} \subset \mathbb{R}^n$ sampled from an unknown submanifold $\mathcal{X}$ in Euclidean space $\mathbb{R}^n$ with intrinsic dimensionality $dim\mathcal{X} = d < n$ (usually $d << n$ is to be expected), find an embedding mapping
>
> $$\mathbf{h} : \mathcal{X} \mapsto \mathbb{R}^d, \quad \mathbf{h}(\mathbf{x}^i) = \mathbf{z}^i \in \mathbb{R}^d \tag{12}$$
>
> which preserves the geometry of the data set X as much as possible, so that the low dimensional data set $\mathbf{Z} = \left[\mathbf{z}^1, ..., \mathbf{z}^m\right]$ is a good representation of the high-dimensional data set $\mathbf{X}$ [44].

Thus, the objective of manifold learning methods is to identify the low-dimensional structures in high-dimensional data such that the behavior of the high-dimensional data can be efficiently analyzed [26]. When the underlying behavior of a problem is sufficiently nonlinear, these algorithms can potentially outperform the previously discussed linear methods [45].

There are many different types of manifold learning algorithms that have been developed in the literature. Each of these methods seeks to solve the same fundamental problem, but primarily differ in the metrics used to obtain the "best" low-dimensional embedding through optimization [46]. Given a set of snapshots for a numerical simulation, manifold learning algorithms generally follow the following broad steps, according to a survey by Izenman [47]:

1) Construct a k-Nearest-Neighbors (kNN) graph with the data points as vertices using neighborhood information around each point
2) Transform graph into a form that is suitable for the next step using a procedure unique to the algorithm
3) Perform a spectral embedding by solving an eigenproblem

Each algorithm yields an embedding $\mathbf{Z} = \left[\mathbf{z}^1, ..., \mathbf{z}^m\right]$ where $\mathbf{z}^i \in \mathbb{R}^d$ and is the low-dimensional representation of the full order snapshot $\mathbf{x}^i \in \mathbb{R}^n$ [26]. A simple illustration of this is shown in Figure 1, which is obtained by applying manifold learning (ISOMAP [45] in this example) to a "Swiss roll" test problem. This example shows data that appear 3-D, as shown in Figure 1a. However, the data exhibit intrinsically 2-D behavior, which is successfully recovered by the manifold learning algorithm in Figure 1b.



(a) Original Data

(b) Low-Dimensional Representation

Fig. 1 Application of manifold learning to Swiss roll test problem

In general, manifold learning problems are not well-posed because the intrinsic dimensionality, d, of the problem is not necessarily known a priori. However, when applied to a data set generated by varying a set of parameters, $\mathbf{p}$, whose dimension is known, it is assumed that the intrinsic dimensionality of the problem is equal to the number of parameters in $\mathbf{p}$ [26]. For this study, ISOMAP [45] is identified as a representative nonlinear DR method and implemented without loss of generality.

D. Interpolation/Regression

Each of the DR techniques discussed in the previous subsection ultimately produces a projection matrix, $\mathbf{Z}$, in which each column gives the coordinates of the corresponding training sample in the low-dimensional latent space. The next step is to obtain a parametric mapping $g_i : \mathbf{p} \mapsto \mathbf{z}^i$ that can be used to estimate the coordinates of unsampled points in the latent space that correspond to unseen independent parameter values. To achieve this, an interpolation or regression model can be trained using pairs of design variables and their corresponding coordinates in the latent space. Cubic splines [48], Radial Basis Functions (RBFs) [19, 20], Gaussian Processes [8, 49], Kriging [50], and Artificial Neural Networks (ANNs) [51, 52] are popular choices for constructing the map between the independent parameters and the latent space. In this study, regressors are trained using an ordinary Kriging formulation with a Mátern 3/2 kernel optimized using the maximum log-likelihood criterion [53].

E. Back-mapping

After the latent space coordinates of an unsampled design, $\mathbf{p}^*$ are estimated through interpolation or regression, the high-dimensional field is estimated through a process called *back-mapping*. For POD, back-mapping is carried out analytically by the projection $\mathbf{x}(\mathbf{p}^*) = \mathbf{\Phi z}(\mathbf{p}^*)$. For manifold learning, an analytical form for back-mapping does not exist, but can be carried out using a variety of methods. Artificial neural networks [51] and Kriging models [51] could be used to carry out this procedure, though the high-dimension of the field can result in poor performance and ill-conditioning. Another method for manifold back-mapping based on Locally Linear Embedding (LLE) [54] has been used in previous studies [19, 20] and is presented below.

For a test design $\mathbf{p}^*$, the interpolation or regression model is evaluated to estimate the latent space coordinates $\mathbf{z}^* = \mathbf{z}(\mathbf{p})^*$. Next, a k-nearest-neighbor method can be used to identify the k snapshots whose low-dimensional projections, $\mathbf{z}^j$, are closest to $\mathbf{z}^*$ in the Euclidean sense. By assuming that the geometry of the recovered manifold is locally linear, it is possible to compute a set of weight coefficients, w_j that approximate $\mathbf{z}^*$ as a linear combination of its k nearest neighbors along the manifold, such that $\mathbf{z}^* \approx \sum_{j=1}^{k} w_j \mathbf{z}^j$ [20, 55]. The values of the weight coefficients, w_j, can be computed by solving a least squares optimization subject to the constraint that $\sum_{j=1}^{k} w_j = 1$. The constraint ensures that the approximation is invariant under linear translation [20]. However, if the number of nearest neighbors is greater than the dimension of the manifold, then the constrained least squares problem does not have a unique solution. To address this issue, a method has been presented by Franz based on work by Saul and Roweis which adds the following terms to the optimization problem [26, 55]:

$$||w||_c^2 = \sum_{j=1}^{k} c_j w_j^2 \tag{13}$$

$$c_j = \epsilon \left(\frac{||\mathbf{z}^* - \mathbf{z}^j||_2}{max_i ||\mathbf{z}^* - \mathbf{z}^i||_2} \right)^{\gamma} \tag{14}$$

Where ϵ and γ are user-defined tuning parameters chosen such that $0 < \epsilon \ll 1$ and $1 < \gamma \in \mathbb{N}$. The quantity c_j is effectively a penalty term applied to the j^{th} snapshot that works to both regularize the optimization and reduce the contribution that snapshots far from $\mathbf{z}^*$ make to its prediction. The result is the following optimization problem, which has a unique solution:

$$\min_{w} ||\mathbf{z}^* - \sum_{j=1}^{k} w_j \mathbf{z}^j||_2^2 + ||w||_c^2 \tag{15}$$

$$s.t. \sum_{j=1}^{k} w_j = 1 \tag{16}$$

The weight coefficients obtained by solving Equations (15)-(16) can be used to reconstruct the high dimensional data as a linear combination of the corresponding high-dimensional snapshots, according to the equation:

$$\mathbf{x}^* = \sum_{j=1}^{k} w_j^* \mathbf{x}^j \tag{17}$$

where $\mathbf{x}^*$ is the reconstructed high-dimensional data at $\mathbf{p}^*$, w_j^* are the weight coefficients that solve Equations (15)-(16), and $\mathbf{x}^j$ are the high-dimensional snapshots that correspond to the k nearest neighbors of $\mathbf{z}^*$in the recovered manifold. The result of this process is a parametric, regression-based ROM that allows the user to predict field data at an arbitrary parameter point $\mathbf{p}^*$ that lies within the domain sampled by the training data.

III. Methodology

The current section describes the DD strategy and subsequent synthesis of the above techniques for reduced-order modeling. In general, the process of creating any ROM is divided into two steps: the offline step where the model is constructed and the online step where the model is evaluated for unseen parameters. In the offline phase, the main steps are:

1) *Identify design space*: Identify a set of b design variables and ranges, $\mathbf{p}, \mathbf{p}_{\min}, \mathbf{p}_{\max} \in \mathbb{R}^b$.
2) *Generate high-fidelity data*: Sample m combinations of design variables, $\mathbf{P} = [\mathbf{p}^1, ..., \mathbf{p}^m] \in \mathbb{R}^{b\times m}$, using a Design of Experiments (DoE) and evaluate the high-fidelity model to generate the solutions $\mathbf{X} = [\mathbf{x}^1, ..., \mathbf{x}^m] \in \mathbb{R}^{n\times m}$.
3) *Decompose the domain and partition data*: Divide the computational domain Ω into S sub-regions, $\Omega_q, q \in \{1, 2, \ldots, S\}$. Each point in Ω is uniquely assigned to only one sub-region, which enables parallel computation during training since there are no overlapping points. In general, the number of sub-regions would be determined by the number of discontinuities in the domain, with each isolated to one sub-region. Further details about the DD procedure applied to each test case are provided later. The high-fidelity solution set is partitioned as follows:

$$\mathbf{X} = \begin{bmatrix} \mathbf{x_1^1} & \mathbf{x_1^2} & \cdots & \mathbf{x_1^m} \\ \mathbf{x_2^1} & \mathbf{x_2^2} & \cdots & \mathbf{x_2^m} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{x_S^1} & \mathbf{x_S^2} & \cdots & \mathbf{x_S^m} \end{bmatrix} = \begin{bmatrix} \mathbf{X_1} \\ \mathbf{X_2} \\ \vdots \\ \mathbf{X_S} \end{bmatrix} \tag{18}$$

where $\mathbf{x_S^1} \in \mathbb{R}^{m_S}$.
4) *Dimensionality Reduction*: Perform Dimensionality Reduction (DR) to project $\mathbf{X_q}$ into their latent space coordinates $\mathbf{Z_q} = [\mathbf{z_q^1}, ..., \mathbf{z_q^m}] \in \mathbb{R}^{d_q\times m}$. If POD is used to carry out DR, then d_q is computed based on the desired RIC threshold, δ, such that $RIC(d_q) \geq \delta$. If manifold learning is used, then $d_q = b$. The proposed technique uses manifold learning in regions where discontinuities are present and POD everywhere else, but DDROMs with only linear DR or nonlinear DR in all sub-domains are also trained for benchmarking.
5) *Fit interpolation/regression model*: Train an interpolation or regression model $\mathbf{g_q}$ such that $\mathbf{g_{q}}_i : \mathbf{p} \mapsto \mathbf{z_{q}}_i$.
6) *Get global POD basis*: Using the high-fidelity solution set, $\mathbf{X}$, generate the global POD basis which will be used for the Gappy-POD procedure.

In the online phase, the process is as follows:

7) *Predict latent variable*: For a given test point, $\mathbf{p}^*$, evaluate $\mathbf{g_q}$ to predict the corresponding latent coordinates, $\mathbf{z_q}^*$ in each sub-region.
8) *Predict high-fidelity reconstruction*: Use the appropriate back-mapping technique to obtain $\mathbf{x_q^*}$. If POD is used, then back-mapping is done by linear projection. If manifold learning is used, then back-mapping can be done by carrying out the locally linear back-mapping outlined in Section II.E.
9) *Generate gappy vector*: Concatenate the vectors in each sub-region to form $\mathbf{x}^*$. For each sub-region interface, the points between the boundary and k^{th} node away from the boundary in sub-domains not containing shocks are masked. This creates a high-dimensional vector with missing information.
10) *Use Gappy-POD to infer missing data*: The gappy data problem is solved to compute the missing elements in the gappy vector and generate the reconstructed solution.

A. Error Metrics

The predictive accuracy of the DDROM is evaluated using multiple quantitative metrics across the set of validation cases, i.e., not used during the training of the model. First, the accuracy of the prediction over the entire field is calculated using the following equation:

$$e_t(\mathbf{x}^{val}, \mathbf{x}^*) = \frac{||\mathbf{x}^{val} - \mathbf{x}^*||_1}{||\mathbf{x}^{val}||_1} \tag{19}$$

where $\mathbf{x}^{val}$ is the true solution at a validation point, $\mathbf{x}^*$ is the predicted solution, and $\|\cdot\|_1$ denotes the L_1 norm.

To better isolate the accuracy of the ROM near the vicinity of the shock, and reduce the impact of errors upstream or downstream of the shock region, the shock error is calculated as follows:

$$e_s(\mathbf{x}^{val}, \mathbf{x}^*) = \frac{\sum_i^v |\mathbf{x}_i^{val} - \mathbf{x}_i^*|}{\sum_i^v |\mathbf{x}_i^{val}|} \tag{20}$$

where $\mathbf{x}_i^*$ is the i^{th} component of $\mathbf{x}^*$ and the sum is evaluated over a set of nodes that are in the vicinity of the shock. Following the procedure implemented in [20] and [19], an engineering approximation is used to infer this set. In particular, this set consists of those nodes whose predicted values deviate from the true values by more than some threshold, τ. For this study, the threshold $\tau = 0.01(\max_i \mathbf{x}_i - \min_i \mathbf{x}_i)$ is used. Stated formally, the set of node indices for a ROM method k, $\mathcal{I}_k$, that are considered to be in the vicinity of the shock is defined as $\mathcal{I}_k := \{i| \, |x_i^{val} - x_i^*| > \tau\}$. To ensure that a holistic quantitative assessment is performed, it is important to note that $\mathcal{I}$ must be the same when comparing different methods. Thus, $\mathcal{I} = \{\mathcal{I}_1 \cup \mathcal{I}_2 \cup \cdots \cup \mathcal{I}_k\}$. More specifically, this is done so that methods that have few indices in their index set do not appear to have a higher error simply because Equation 20 is evaluated over fewer components. Nodes that are not part of the shock index set are grouped into a non-shock index set, which allowed the error in the regions away from the shock to assessed separately. In the results to follow, the nodes (not) in the shock index set are referred to as (non-)shock nodes; the error evaluated over these nodes is called (non-)shock error.

The global and shock error calculation can both be decomposed further to sub-domain specific errors. In particular, by restricting the index set to just the nodes in each sub-domain, the above metrics can be used to evaluate the performance of each local DDROM. Similarly, by computing the error over only those nodes that are *not* at the domain interfaces (where Gappy-POD is applied), the impact of DD and Gappy-POD on the predicted solutions can be individually assessed. This enables a more detailed analysis and prevents the performance of Gappy-POD at sub-domain interfaces from confounding the DD results in the rest of the field.

Recognizing that each non-intrusive DDROM is essentially a combination of a dimensionality reduction model and a regression model, the global error can be alternatively decomposed into a reconstruction and regression error. The reconstruction error, e_{rc}, is the error associated with how well the latent space models the physical space. More specifically, because nonlinear DR results an implicit map from the physical space to the latent space, the reconstruction error can only be defined for the linear ROMs, where it represents the accuracy of the POD modes. The error is given by

$$e_{rc}(\mathbf{x}^{val}, \mathbf{\Phi}) = \frac{\|\mathbf{x}^{val} - \mathbf{\Phi}\mathbf{\Phi}^T\mathbf{x}^{val}\|_1}{\|\mathbf{x}^{val}\|_1} \tag{21}$$

The regression error, e_{rg}, quantifies the impact of discrepancies in the latent space coordinates caused the latent space interpolation models. This is given by

$$e_{rg}(\mathbf{x}^{val}, \mathbf{\Phi}) = \frac{\|\mathbf{\Phi}(\mathbf{\Phi}^T\mathbf{x}^{val} - \tilde{\mathbf{z}})\|_1}{\|\mathbf{x}^{val}\|_1} \tag{22}$$

When the regression and reconstruction error are computed in each sub-domain, only those components of $\mathbf{\Phi}$ and $\mathbf{x}^{val}$ that belong to the sub-domain are used in the calculations.

IV. Experimentation and Results

The methodology proposed in this study is characterized using three test cases that progressively increase in complexity. First, an experiment is performed on nonlinear quasi-1D flow through a converging-diverging channel. This is followed by an analytical test problem resembling supersonic flow over a 2D wedge. The final experiment considers transonic flow over an RAE2822 airfoil. The objectives of the experiments are: 1) characterize the performance of DD for each DR method; 2) assess any improvements obtained by combining DR methods after the domain is partitioned; 3) evaluate the Gappy-POD method for reducing gradients between sub-domains.

To begin each test problem, benchmark ROMs are trained without applying DD to illustrate the predictive capabilities of current methods. Global ROMs are trained using both linear and nonlinear methods, and are referred to as Linear Global (LG) and Nonlinear Global (NLG) respectively for this section. In this study, POD is used as the representative linear ROM method while ISOMAP is used as the representative nonlinear ROM method. Next, ROMs are trained using the proposed DD approach. To characterize the effects of DD, different combinations of ROM techniques are applied across each sub-domain. For this study, DDROMs using only linear local models (referred to as LDD), only nonlinear local models (referred to as NLDD), and a mix of both nonlinear and linear local models (referred to as MDD) are considered. For all DD-based models, domains are subdivided such that discontinuities are contained within a single sub-domain. When training MDD ROMs, nonlinear methods are used in the sub-domain containing the discontinuity while linear models are used in the other sub-domains. Table 3 summarizes the different ROMs and their attributes.

Table 3 Different types of ROMs generated for each test case.

Acronym	Type	DR Method	Interface
LG	Global ROM	POD Only	N/A
NLG	Global ROM	ISOMAP Only	N/A
MDD	DDROM	ISOMAP + POD	Gappy-POD
LDD	DDROM	POD Only	Gappy-POD
NLDD	DDROM	ISOMAP Only	Gappy-POD

A. Test Case 1: Quasi-1D Flow through Converging-Diverging Nozzle

The governing equations for steady flow through a converging-diverging nozzle are given below.

$$\begin{aligned}\frac{\partial(\rho u A)}{\partial x} &= 0\\ \frac{\partial(\rho u^2 A + \frac{pA}{\gamma})}{\partial x} - p\gamma\frac{\partial A}{\partial x} &= 0\end{aligned} \tag{23}$$

where $A = A(x)$ is the area distribution, ρ is density, p is pressure, u is velocity, and γ is the specific heat ratio of the fluid. By specifying the boundary conditions, length of the nozzle, L, and area distribution, one can iteratively solve for the Mach number, pressure, and density inside the nozzle [56]. In this study, the exit area to throat area ratio (A_e/A_t) is varied and the pressure ratio throughout the nozzle is predicted. The area profile of the nozzle parabolically changes from $A(-L/2) = A(L/2) = 3$, with the throat located at $x = 0$. The length is set to 10 units. The boundary conditions, $p(-L/2) = 1$, and $p(L/2) = 0.7$, are chosen such that the shock forms in the diverging portion of the nozzle. The solution is obtained by discretizing the governing equations over a uniform grid of 1000 points and using a bisection method to find the shock location. A LHS DOE is used to sample $A_e/A_t \in [2, 8]$ with 75 samples used for training and 25 for validation. Each ROM is trained 200 times using different random seeds and the error metrics are averaged over all replications.

Figure 2 illustrates the sub-domains used to train DDROMs for the test case. These sub-domains were obtained by computing the shock location for all points within a given sampling and setting the sub-domain bounds such that the shock lies within the same sub-domain across all samples. In the MDD case, ISOMAP is applied in the sub-domain containing the normal shock while POD is applied in the remaining two sub-domains. For LDD and NLDD, the sub-domains remained the same, but the same DR method was applied within all sub-domains.

Figure 3 shows the predicted pressure variation for LG, NLG, LDD, and MDD. As the pressure prior to the shockwave is relatively insensitive to the throat area, all the ROMs are able to accurately capture the pressure changes with little error. The regions surrounding the shock (Sub-Domain 2) and after the shock (Sub-Domain 3) are more difficult to model as the step location and size are highly sensitive to the throat area. While all ROMs are able to capture the location of the step, the predicted solution from LG and LDD exhibit spurious oscillations which are highly undesirable. However, methods which applied nonlinear DR in Sub-Domain 2 (namely NLG, MDD, and NLDD) do not exhibit such oscillations and thus are able to obtain more accurate predictions overall. Table 4 shows the error calculated over each sub-domain and globally. While all ROMs are accurate on this relatively simple problem, some general trends that repeat in other test cases can be seen: 1) ROMs developed using the ISOMAP method have approximately a 30% reduction in error in the shock region compared to linear ROMs due to the lack of oscillations; 2) domain decomposition reduces the non-shock error (Sub-Domain 1 and 3) for NLDD compared to NLG; 3) by combining DR methods, MDD is able to simultaneously reduce error in all regions. MDD and NLDD are the most accurate methods tested in this problem due to their accuracy near the shock, which accounts for most of the overall error in this case.

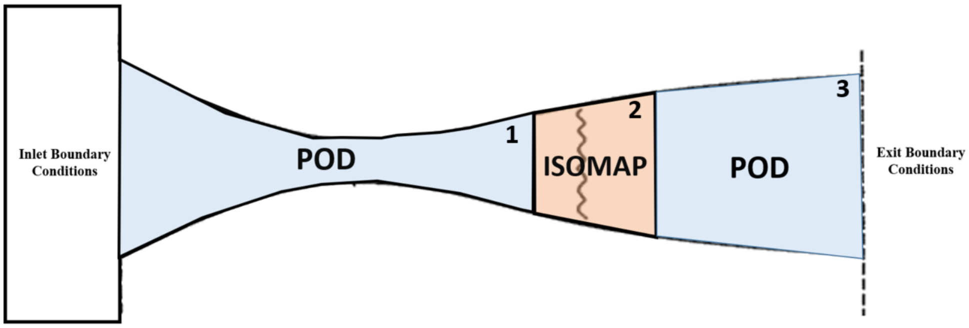


Fig. 2 CD Nozzle: Sub-domains used in training MDD ROM.

Table 4 CD Nozzle - Total prediction error for different ROMs in each sub-domain.

ROM	Sub-Domain 1 [10^{-4}]	Sub-Domain 2 [10^{-4}]	Sub-Domain 3 [10^{-4}]	Full Domain [10^{-4}]
LG	1.63	220.0	10.3	29.5
NLG	3.33	153.0	10.7	22.1
MDD	2.55	154.0	9.24	21.4
LDD	2.54	218.0	9.25	29.8
NLDD	1.35	154.0	9.84	20.7

B. Test Case 2: Flow over a Wedge with Shocks

The next test problem examined in this study is an analytical step function over a 2-D domain that is meant to resemble an attached oblique shock that may form in supersonic flow over a wedge. Using analytical functions allows for more direct control over the features of interest. The spatial domain is discretized using a 200 × 200 grid and the field response that is predicted in this test problem is given by the following governing equation:

$$f(x, y; a, \beta, \delta) = \begin{cases} a^2 + b, & \text{if } y \leq x \tan(\beta + \delta) \\ b, & y > x \tan(\beta + \delta) \end{cases} \tag{24}$$

where a and b determine the shock step size, δ is the wedge angle, and β is the shock angle relative to the wedge surface. In this problem, the wedge angle is fixed to 10°, while all other inputs are varied in the ranges shown in Table 5. Each model is trained using a set of 200 samples and validated on 50 samples, both generated using an LHS design of experiments. The above experiment is repeated for 200 different DOEs, each generated using a different random seed, and the results are averaged over the runs.

Table 5 2D Wedge - Input parameter ranges used to generate the LHS DOE.

Parameter	Min	Max
a	1	11
b	5	21
β	5°	20°

Similar to the converging diverging nozzle test case, the sub-domains were chosen by analyzing the shock location at every snapshot. Sub-Domain 2 is defined such that the step is at least 5 nodes away from the sub-domain boundary. The Gappy-POD domain is selected such that any gradients between sub-domains are smoothed out and is at least 10 nodes wide. The sub-domains used for the DDROMs and the Gappy-POD region are shown in Figure 4. Table 6 presents the

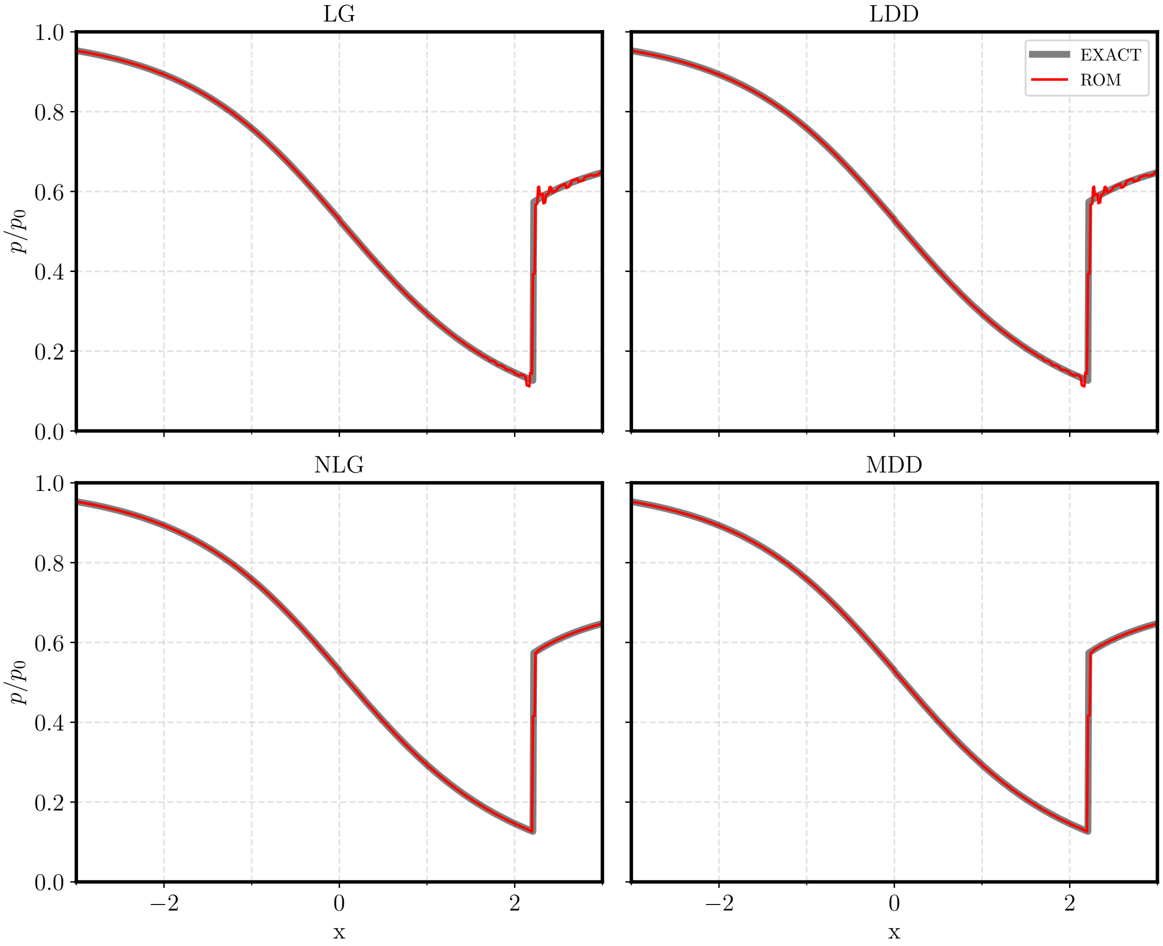


Fig. 3 CD Nozzle - Impact of DR method on the predicted pressure variation through the nozzle.

errors for each of the ROMs evaluated over each individual sub-domain and the full domain. As was observed in the previous test problem, NLG models have a 33% reduction in the shock error compared to LG models. For the non-shock nodes, DD enables a 9% reduction in error for NLDD compared to NLG. While DD does not improve the performance of NLDD models near the shock, there is a 6% reduction seen for LDD models when compared to LG models. Due to their accuracy near the shock and the outsized effect that shock prediction has on total accuracy, both MDD and NLDD are equally effective in predicting the outputs, outperforming LG models by 46% in total error. It is important to note that all domain decomposition based ROMs outperformed their global counterparts, with LDD models seeing a 9% improvement over LG models and NLDD models seeing a 6% improvement over NLG models in total error.

The selection of a given ROM technique not only depends on numerous error metrics for quantifying performance, but also on its ability to capture the underlying structure of the data. In Sub-Domains 1 and 3, all DDROMs have at most a latent space dimensionality of 3; meanwhile, LG has a latent space dimensionality of 111. Table 7 shows the regression and reconstruction errors for the ROMs generated using linear DR. It is clear that DD not only creates a simpler model requiring fewer modes, but that the modes in each sub-domain are better aligned with the data since the reconstruction error is several orders of magnitude lower. Due to the lower latent space dimension, the regression error of the surrogates used to predict latent space coordinates, are also orders of magnitude more accurate. Such a reconstruction and regression error breakdown cannot be performed for nonlinear ROMs due to the lack of an explicit mapping from the high dimensional space to latent space. However, it can be deduced that the latent space identified by the DDROMs is superior as total error is reduced while the latent space dimensionality remains the same.

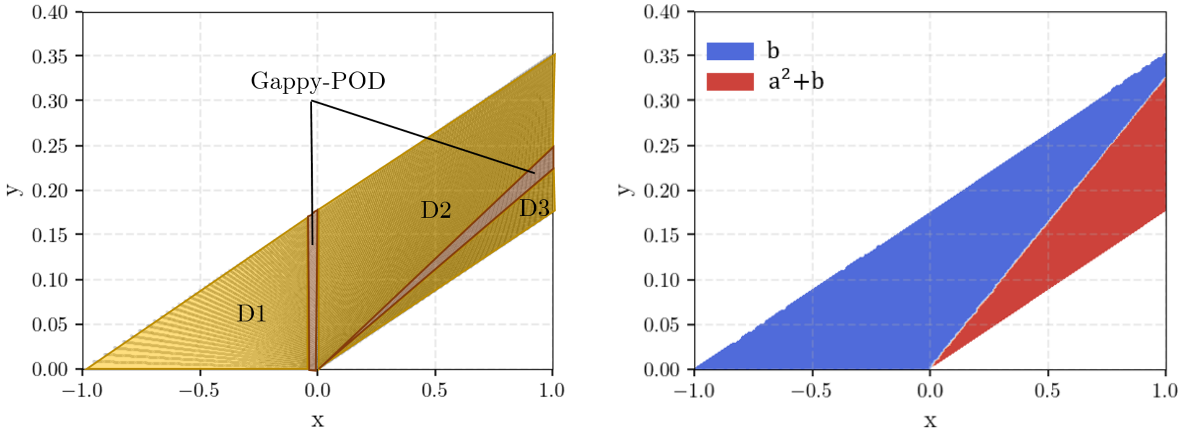


Fig. 4 2D Wedge - Left: Sub-Domains used in training DDROMs where "DX" means Sub-Domain X. Right: Exact field solution with $a = 8$, $b = 5$, and $\beta = 5°$. Note that the leading edge of the wedge is located at (0,0).

Table 6 2D Wedge - Prediction error for different ROMs evaluated around and away from the shock.

ROM	Non-Shock Nodes [10^{-3}]	Shock Nodes [10^{-3}]	Full Domain [10^{-3}]
LG	1.91	90.2	12.0
NLG	3.79	60.6	6.87
MDD	3.45	61.7	6.44
LDD	4.20	84.6	10.9
NLDD	3.43	61.8	6.44

C. Test Case 3: Transonic Flow over an Airfoil

The goal of this test case is to assess the ability of the proposed methodology to predict the pressure field around a transonic airfoil with variable shape and angle of attack. The data set used in this study is identical to that used in a previous study by Perron et al. [15], a brief discussion of which is presented in this section. The RAE2822 airfoil is parameterized using a Free Form Deformation (FFD) approach as illustrated in Figure 5 [57]. The airfoil is enclosed in a bounding box with boundary nodes dictating the degrees of freedom available for parameterization. The shape is varied by vertically displacing interior control points while holding the control points at the leading and trailing edges fixed to preserve angle of attack. To examine the impact of input space dimension on model performance, this problem is repeated by varying the number of Design Variables (DVs) from 3 to 15. Control point displacement is limited to ±3% of the chord length, while angle of attack is allowed to vary from 0° and 4°. For each case, the freestream Mach number is fixed at $M_\infty = 0.725$ such that transonic effects are observed. The simulations are performed using the

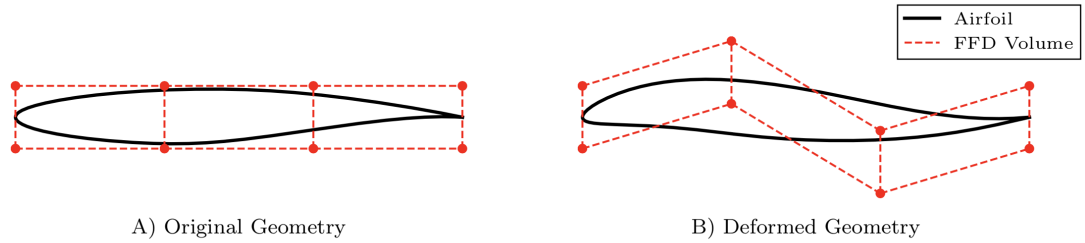


Fig. 5 Notional shape perturbation using FFD

Table 7 2D Wedge - Comparison of reconstruction and regression errors for linear ROMs.

ROM	Metric	Sub-Domain 1	Sub-Domain 2	Sub-Domain 3
LG	e_{rc}	2.74×10^{-3}	5.90×10^{-3}	5.47×10^{-4}
	e_{rg}	1.16×10^{-3}	4.25×10^{-3}	2.29×10^{-4}
LDD	e_{rc}	5.77×10^{-15}	5.70×10^{-3}	1.29×10^{-14}
	e_{rg}	9.10×10^{-15}	4.23×10^{-3}	3.36×10^{-5}

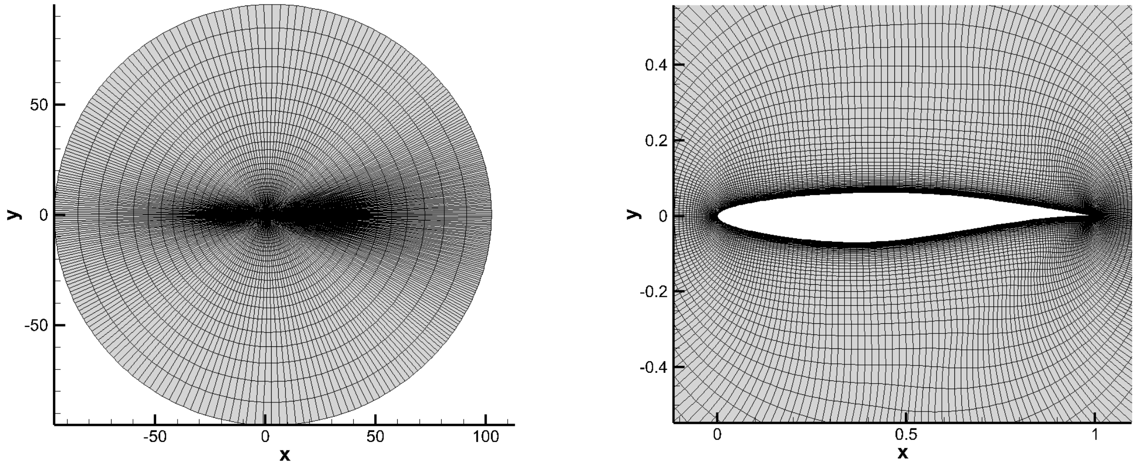


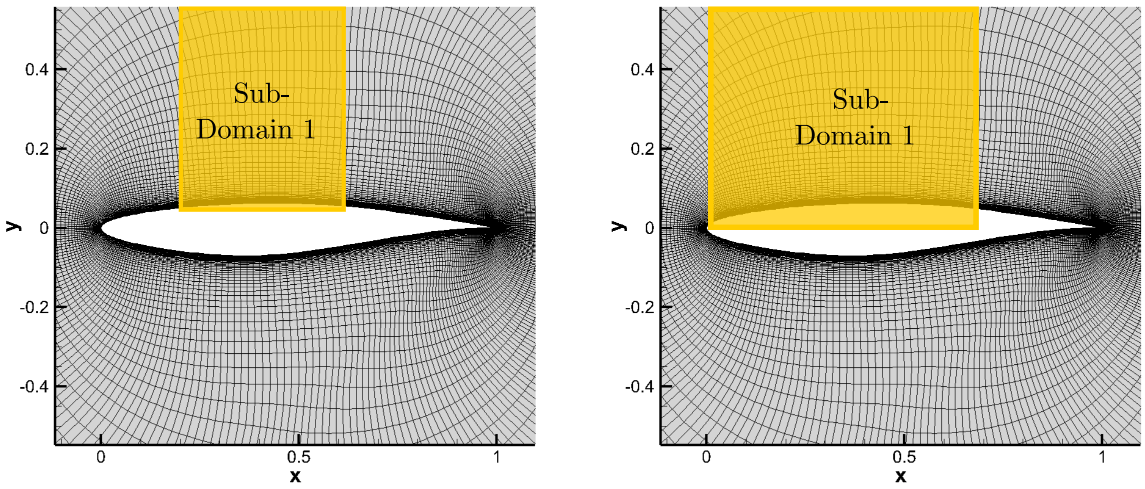
Fig. 6 RAE2822 - CFD grid

open-source SU2 code [58], which provides the facilities to generate new shapes and update the grid suitably. Solutions are computed using Reynolds-Averaged Navier-Stokes (RANS) CFD with a Spalart-Allmaras (SA) [59] turbulence model. The computational domain is discretized using a structured O-grid topology, as shown in Figure 6, with 10,500 nodes. A database of 2,500 samples are generated for each number of DVs using an LHS. Models are trained using 500 randomly selected samples from this database, with an additional 50 samples being randomly selected for validation. The above experiment is repeated using 100 different random seeds and the results are averaged over the runs.

Remark: In the previous experiments, gradients between sub-domains were extremely small, and hence the predictions were insensitive to the Gappy-POD algorithm applied at the interfaces. However, for this test problem, more complex flow features result in the presence of artificial gradients at sub-domain interfaces. To isolate and individually assess the impact of DD and Gappy-POD on the predicted solutions, this section first presents the results excluding nodes where Gappy-POD interpolation was applied. Then, the Gappy-POD method's performance is separately evaluated by including the interface region in the error metrics. This enables a more detailed analysis and prevents the performance of Gappy-POD at sub-domain interfaces from confounding the DD results in the rest of the field.

1. Domain Decomposition Strategy

For each of the DDROMs, the computational domain was split into 2 sub-domains. The first domain is restricted to the upper-surface of the airfoil where a shockwave is present; the second contains the remaining nodes in the field. The nodes in each domain are determined by setting a rectangular boundary and selecting the subset of nodes bounded by that box. As the mesh used in the CFD simulations was not Cartesian, the nodes bounding each domain do not form a rectangle, even though the boundary specified was a rectangle. Previous work using DD has relied on engineering judgement to set the domain boundary. In this study, however, the size of the rectangular bounding box is varied and translated to find the optimal DDROM. More specifically, the boundaries of the sub-domains were initially set by determining the shock's location in a subset of the training data. Then, a gradient free, direct search optimization was performed, which involved incrementally moving each of the boundaries until a minimum total error was found for the DDROM. This is referred to as a compass search optimization algorithm in literature [60]. This procedure was repeated for each DV data set to ensure that the sub-domains identified were optimal for that data set. A three sub-domain decomposition was also investigated, but this did not improve model accuracy and hence is not presented. Further work is needed to increase the efficiency of the DD strategy. In particular, methods for detecting nonlinearity in the computational domain using shock sensors and automatic clustering are being investigated to streamline the offline ROM generation process. A gradient-based optimizer with a multi-start procedure could also improve the domain shape optimization. Figure 7 shows the optimized bounding rectangles for the upper-surface airfoil sub-domain for 3DV and 15DV data sets.



(a) DV = 3

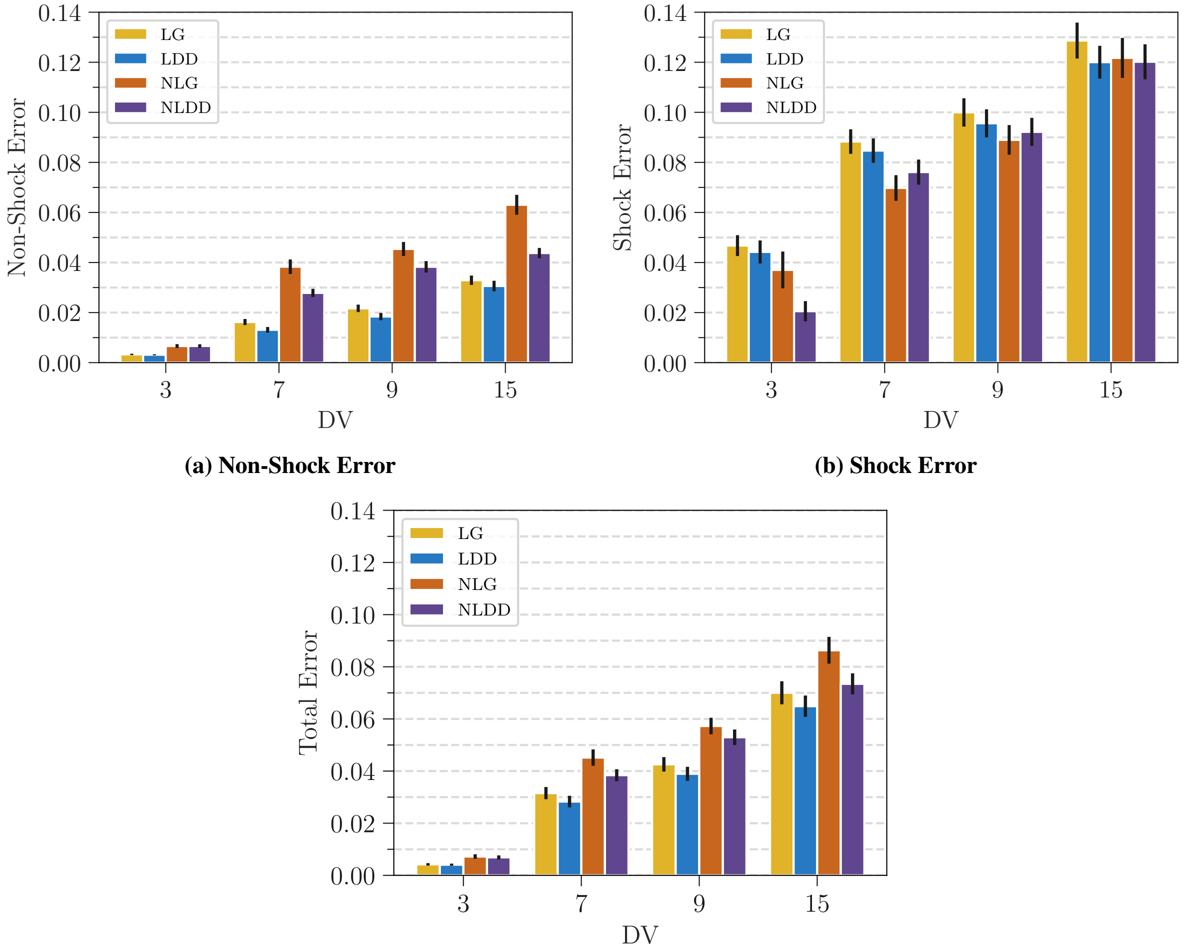
(b) DV = 15

Fig. 7 RAE2822 - Optimized upper surface sub-domain for 3 DV and 15 DV test cases. Note: All points not in Sub-Domain 1 are in Sub-Domain 2.

2. Impact of Domain Decomposition on Linear and Nonlinear ROMs

Figure 8 shows the impact of partitioning the global domain into sub-domains when using only linear or nonlinear DR. As expected, all the error metrics monotonically increase with DV count because the shock location and pressure field have more variability. The superior performance of ISOMAP near the shock and POD away from the shock was seen in the other test cases as well as in previous papers on the RAE2822 airfoil in transonic flow [9, 19]. This is because POD relies on a linear expansion using POD modes; as the relative information criteria increases, more modes and degrees of freedom are required to capture the variability in the model. The later POD modes cause oscillations around the shockwave, most evident in the converging diverging nozzle results, consequently increasing the shock error. On the contrary, ISOMAP's algorithm, particularly its back-mapping step, are heavily regularized and are less vulnerable to exhibiting spurious oscillations at the shock. However, a consequence of this regularization appears to be that nonlinear models exhibit higher error outside of the shock region than their linear counterparts.

ROMs developed using linear DR achieve a 5% lower shock error (on average) when DD is used. In fact, the relative reduction in shock error increases with an increase in the number of design variables. There is a 4% decrease in non-shock error for the 3DV case, a larger 19% decrease for the 7DV, and a 7% decrease for 9DV. These performance improvements in the shock and non-shock cells lead to a total error reduction of 7% for LDD compared to LG on average over all DV data sets. This performance improvement of LDD over LG models may be attributed to the difference in latent space dimensions obtained using each approach. By building a series of local ROMs, each sub-domain must capture a subset of the physics of the entire computational domain. Hence, for a given *RIC*, the number of modes required to capture the dynamics should decrease if the modes are more locally accurate. Indeed, as seen in Table 8, the number of modes in each sub-domain is always less than the number of global modes. Because DDROMs have a lower dimensional latent space, it is likely that the resulting interpolation models are introducing less error into the prediction



(a) Non-Shock Error

(b) Shock Error

(c) Total Error

Fig. 8 RAE2822 - Impact of DD on linear and nonlinear DR methods. The lines on each bar indicate the standard deviation of the data evaluated over all the replications.

than their global counterparts.

Figure 8 also shows that NLDD reduces the non-shock error by 27% for 7DV and 31% for 15DV. For the 3DV case, there is inconsequential difference in the non-shock error. Because the shock is only a small subset of the total domain, the decrease in non-shock error leads to a 10% reduction in total error for NLDD compared to NLG on average over all DV data sets. Hence, DD provides superior performance in total predictive accuracy irrespective of the DR technique used.

3. Evaluation of Combining Dimension Reduction Techniques on DDROM

Table 9 and the accompanying Figure 9 show the trends in predictive performance as the number of DVs is increased. The table quantifies the improvement in accuracy for each of the DDROMs to linear global ROMs. As expected, since LDD uses linear DR in each sub-domain, it has the lowest non-shock error and provides largest reduction compared to LG. Note that unlike the previous test cases, NLDD does not perform the best because the non-shock region is highly sensitive to the shape/design variables. Thus, MDD, which uses linear DR away from the airfoil has a similar performance to LDD and NLDD actually degrades in performance compared to LG. MDD outperforms both NLDD and LDD in shock error, though all ROMs converge to similar errors as the number of DVs increases. As initially hypothesized, by combining both DR methods in the ROM, MDD is able to closely match LDD's performance in the non-shock error and NLDD's performance in the shock region. As MDD has the best attributes of both dimension

Table 8 RAE2822 - Number of modes required by each linear ROM with RIC=99.99% in each sub-domain.

DV	LG	LDD	
		Sub-Domain 1	Sub-Domain 2
3	21	16	12
7	37	29	17
9	43	30	22
15	54	37	27

reduction methods, it performs best in total predictive error. Figure 10 shows reconstructed C_p fields from a 3 DV test case to visualize the qualitative accuracy of LDD compared to MDD. Predictions made by the linear ROMs not only produce larger errors, but also result in an oscillatory error pattern around the shockwave. The nonlinear models, on the other hand, produce a smaller shock error region and do not predict spurious oscillations in the field. It is worth noting that MDD and LDD have very similar predictive accuracy in total error. This may be because MDD provides the greatest improvement in the shock region when compared to linear ROMs, but, even for the case with 15 DVs, the number of shock cells is merely 12% of the entire computational domain. Indeed for the 2D wedge experiment, where the shock is a dominant and largest contributor to the total error, MDD vastly outperforms LDD.

From these results, it may be concluded that MDD is most likely best suited for problems where the number of shock cells and strength of the shock are both large. Results also show that MDD provides a benefit when the number of design variables is small. As the objective of the study was to explore ROM methods that improve the predictions both near and away from shocks, MDD is superior to LDD.

Table 9 RAE2822 - Comparison of the prediction errors for different DDROMs relative to linear global models decomposed into shock, non-shock, and total error.

(a) Non-Shock Error

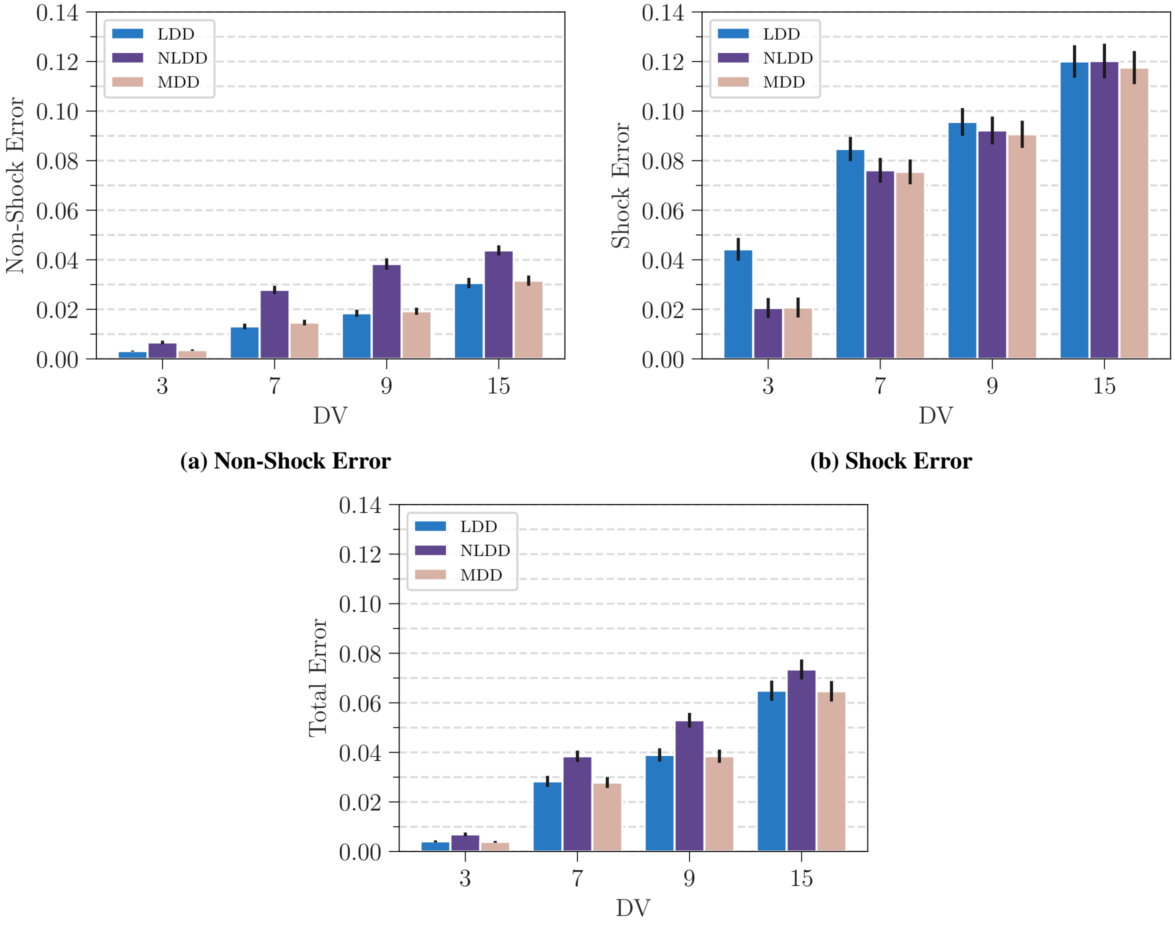
DV	LG [10^{-3}]	Change rel. to LG		
		LDD	MDD	NLDD
3	3.28	-4%	8%	102%
7	16.2	-19%	-10%	72%
9	21.7	-15%	-11%	77%
15	32.9	-7%	-4%	33%

(b) Shock Error

DV	LG [10^{-3}]	Change rel. to LG		
		LDD	MDD	NLDD
3	46.8	-5%	-56%	-56%
7	88.3	-4%	-15%	-14%
9	100.0	-4%	-9%	-8%
15	129.0	-7%	-9%	-7%

(c) Total Error

DV	LG [10^{-3}]	Change rel. to LG		
		LDD	MDD	NLDD
3	4.23	-3%	-8%	63%
7	31.5	-10%	-12%	22%
9	42.6	-9%	-10%	24%
15	70.0	-7%	-8%	5%



(a) Non-Shock Error

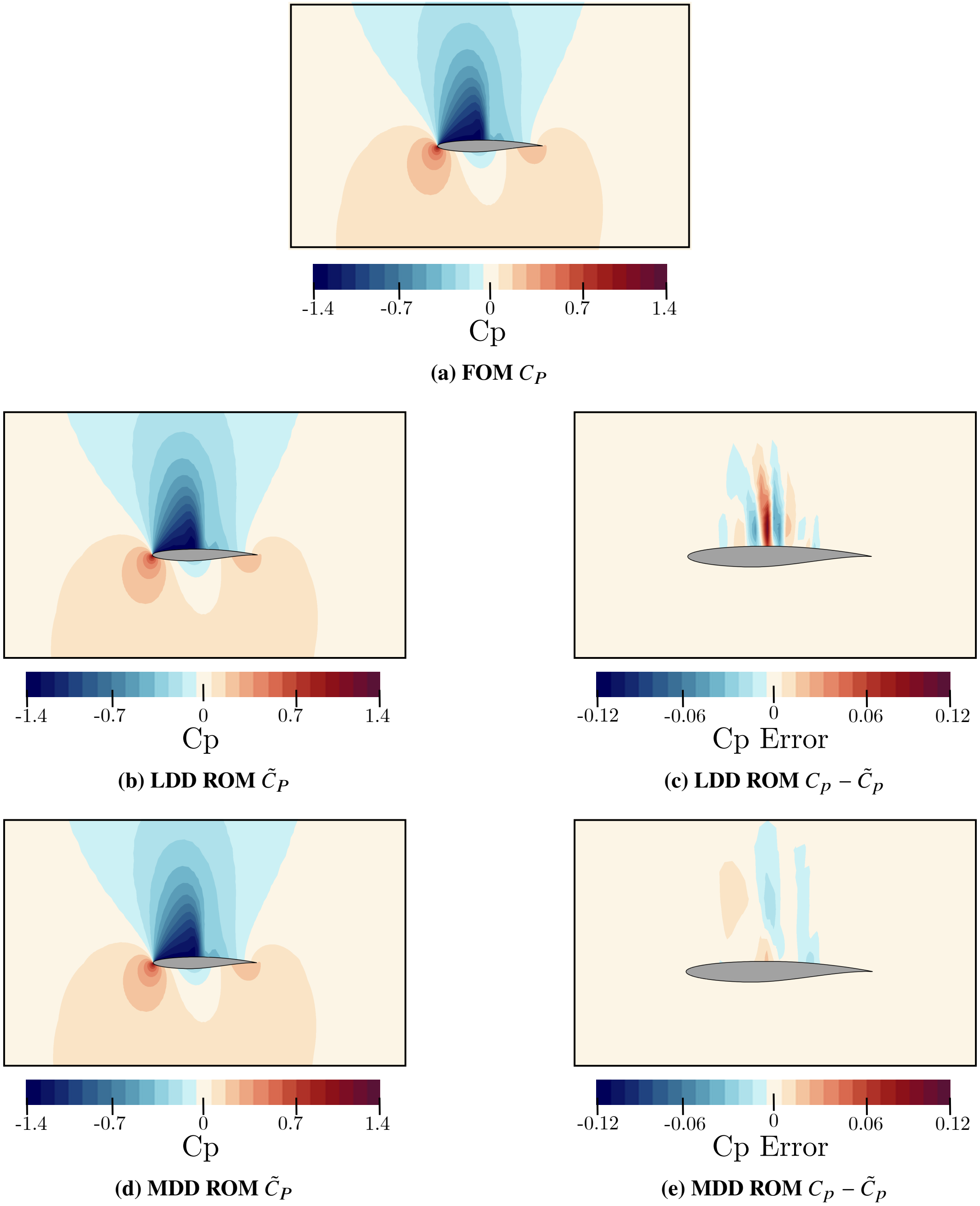
(b) Shock Error

(c) Total Error

Fig. 9 RAE2822 - Comparison of MDD ROM to linear and nonlinear DDROMs. The lines on each bar indicate the standard deviation of the data evaluated over all the replications

4. Impact of Gappy-POD Interface Stitching

Recall that when DDROMs are invoked for predictions, there tend to be sharp gradients that occur between sub-domains. These must be reduced for the solution to be more physical. Table 10 shows the change in total error averaged over 100 replications when the Gappy-POD is used at the domain interfaces to mitigate such gradients. While all ROMs see an increase in error, the linear models have a consistently smaller change when Gappy-POD is used. This is because both Gappy-POD and LDD use the same DR method; thus, the basis used to determine the values in the Gappy-POD domain is likely to be similar to the basis set used in the surrounding domains. On the other hand, nonlinear models such as MDD and NLDD use a completely different DR process and thus obtain a significantly different latent space projection. This leads to spurious values at the interfaces. As NLDD uses ISOMAP in all sub-domains, it is most significantly impacted by Gappy-POD.



(a) FOM C_P

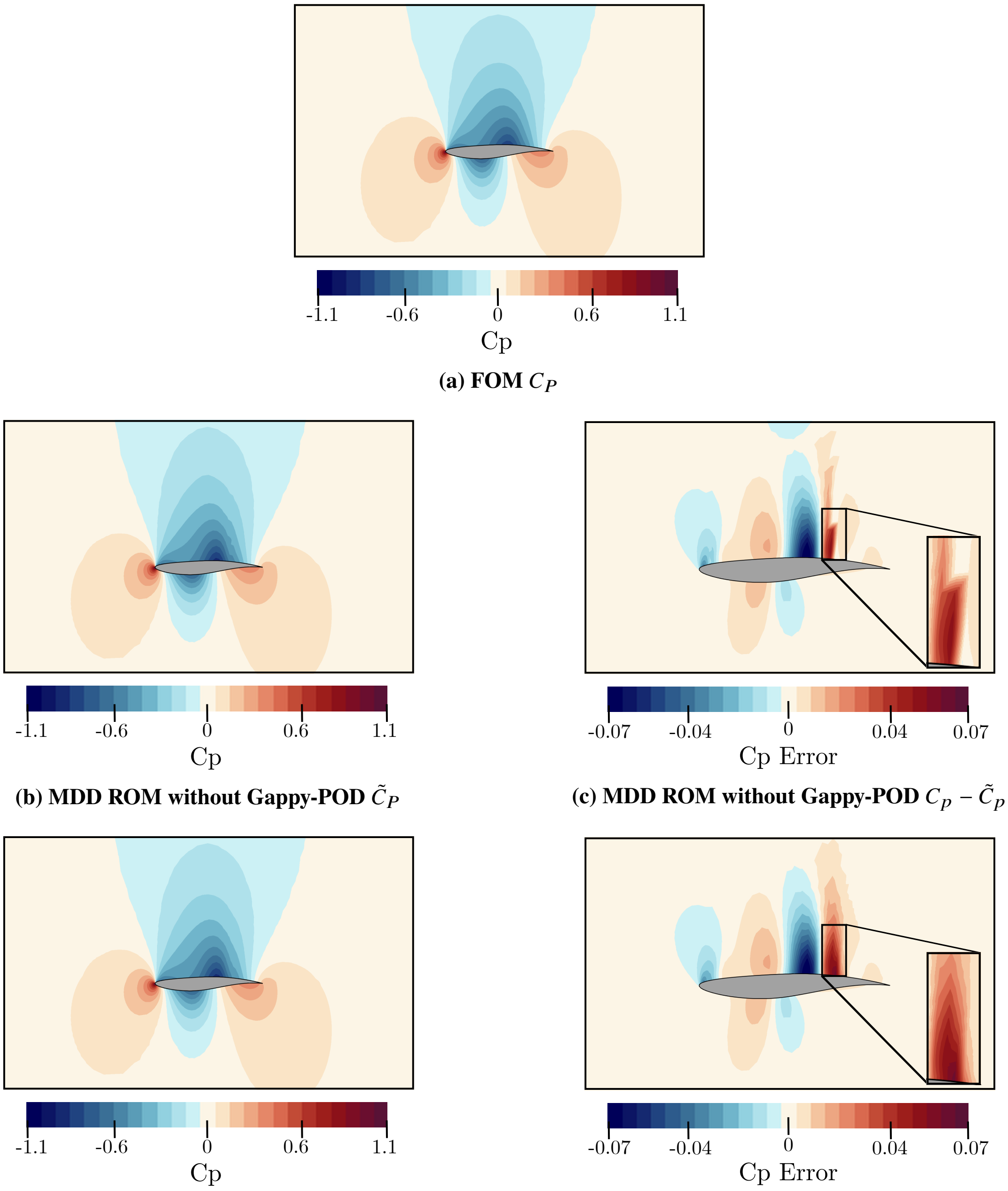
(b) LDD ROM $\tilde{C}_P$

(c) LDD ROM $C_p - \tilde{C}_p$

(d) MDD ROM $\tilde{C}_P$

(e) MDD ROM $C_p - \tilde{C}_p$

Fig. 10 RAE2822 - Flow visualization for the predicted C_P and prediction discrepancy for LDD and MDD ROM. Solutions are for a test case with 3 DVs.



(a) FOM C_P

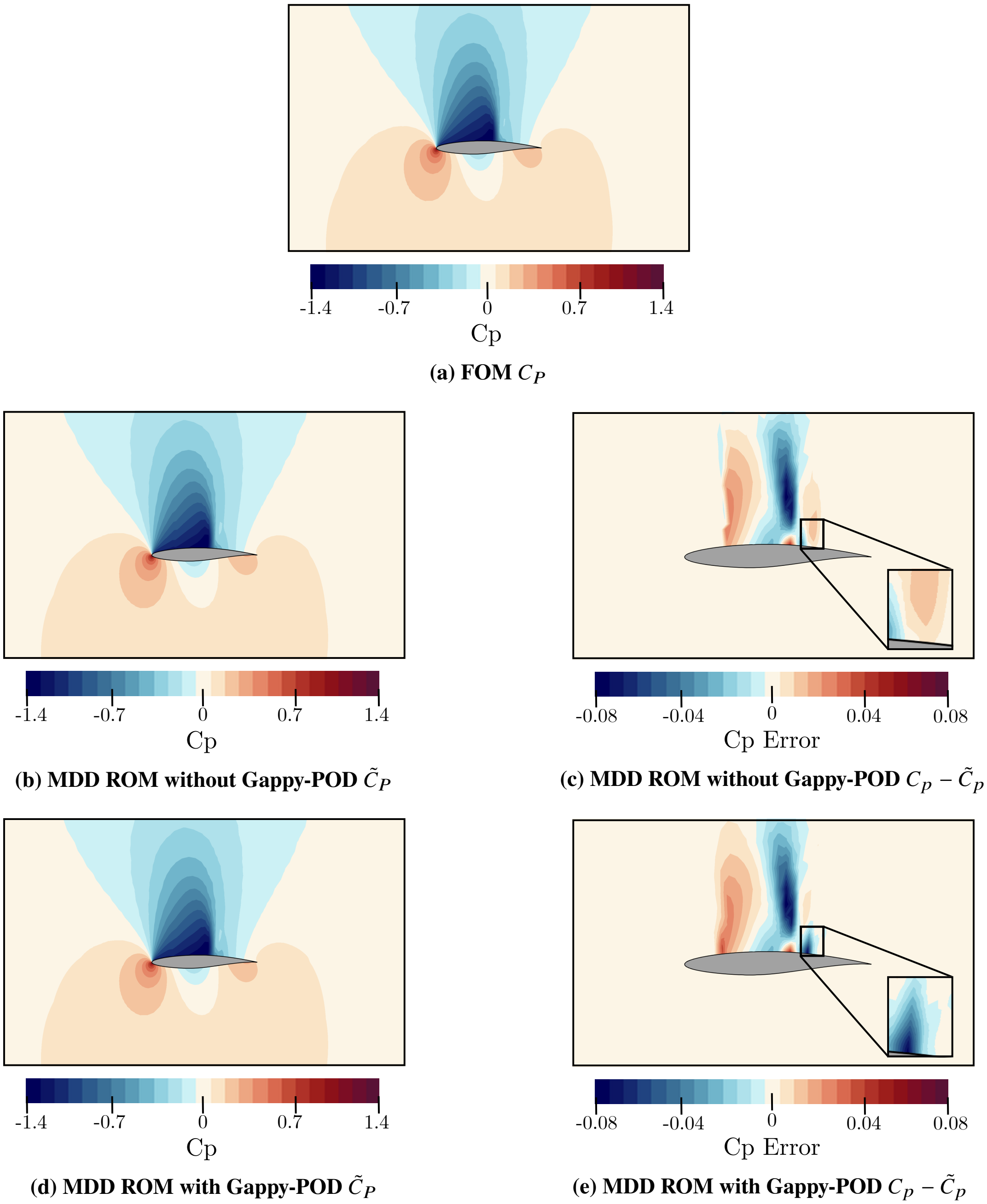
(b) MDD ROM without Gappy-POD $\tilde{C}_P$

(c) MDD ROM without Gappy-POD $C_p - \tilde{C}_p$

(d) MDD ROM with Gappy-POD $\tilde{C}_P$

(e) MDD ROM with Gappy-POD $C_p - \tilde{C}_p$

Fig. 11 RAE2822 - Flow visualization for the predicted C_P and prediction discrepancy for MDD ROM with and without Gappy-POD. Solutions are for a test case with 9 DVs. Note that Gappy-POD successfully reduces the sharp gradient between sub-domains visible in the inset in (c).



(a) FOM C_P

(b) MDD ROM without Gappy-POD $\tilde{C}_P$

(c) MDD ROM without Gappy-POD $C_p - \tilde{C}_p$

(d) MDD ROM with Gappy-POD $\tilde{C}_P$

(e) MDD ROM with Gappy-POD $C_p - \tilde{C}_p$

Fig. 12 RAE2822 - Flow visualization for the predicted C_P and prediction discrepancy for MDD ROM with and without Gappy-POD. Solutions are for a test case with 3 DVs. Note that in this case Gappy-POD introduces additional error visible in the inset in (e).

Table 10 RAE2822 - Impact of Gappy-POD on the total prediction errors for different DDROMs.

DV	ROM	W/O Gappy-POD [10^{-3}]	W/ Gappy-POD [10^{-3}]	Change
3	MDD	3.98	4.84	21.44%
	LDD	4.16	4.76	14.51%
	NLDD	6.91	8.72	26.22%
7	MDD	26.9	27.7	2.87%
	LDD	27.4	27.7	1.06%
	NLDD	37.5	39.3	4.66%
9	MDD	38.2	39.7	4.00%
	LDD	38.7	39.7	2.73%
	NLDD	53.3	59.4	11.58%
15	MDD	64.2	64.6	0.61%
	LDD	64.4	64.6	0.28%
	NLDD	73.3	73.9	0.75%

Figures 11 and 12 show the absolute error values throughout the field and the capabilities of Gappy-POD. In general, the algorithm is able to successfully reduce gradients at interfaces where the gradients and surrounding sub-domain errors are originally small. This is seen in Figure 11 where Gappy-POD produces a qualitative improvement in the predicted field because of the mitigation of gradients. However, most domain interfaces had sharper gradients with larger error in surrounding sub-domains. In such cases, Gappy-POD extends the region of high error into the interface region and increases total error (see Fig. 12). This behavior is most pronounced in the 3 DV test cases because the sub-domain boundaries are closer to the shock, which increases the error in the region immediately surrounding the interface.

V. Conclusion

The aim of this paper was to develop a reduced order modeling method capable of effectively predicting aerodynamic flows with discontinuous flow features, such as shocks. Previous studies had identified nonlinear ROMs as key enablers in predicting high-dimensional flows with such complex flow features, but showed that linear ROMs have higher accuracy in regions of the field that were far from discontinuities. This paper develops a parametric, domain decomposition based reduced-order modeling method that leverages both linear and non-linear dimension reduction.

The results reveal that DD, irrespective of the DR method, helps reduce the total error when predicting fields with discontinuous features. For linear DDROMs, the identified latent space not only better captures the dynamics, but does so with fewer modes than linear global ROMs. Moreover, the error caused by the characteristic spurious oscillations exhibited near shocks was consistently smaller with DD. This benefit increased when experiments had more input parameters and varying shock location. Nonlinear DDROMs, on the other hand, exhibited a clear reduction in error away from shocks and performed as well as global nonlinear ROMs near discontinuities. Models that used a combination of ISOMAP (near the shock) and POD (elsewhere) showed superior performance compared to those that used just POD or ISOMAP in each sub-domain. This performance improvement was seen in all experiments highlighting the robustness of the proposed approach.

In the final test problem, Gappy-POD was shown to reduce the introduction of artificial gradients at sub-domain interfaces. However, if the gradients were particularly large, total error increased with Gappy-POD stitching. Future work will investigate alternate methods for stitching sub-domains or modifications to the Gappy-POD algorithm to improve its performance. Particular importance should be placed on methods that are amenable to stitching sub-domains that use nonlinear DR. Further work is also needed to increase the efficiency of the DD strategy. Currently, methods for detecting nonlinearity in the computational domain using shock sensors and automatic clustering are being investigated to streamline the offline ROM generation process.

References

[1] Ripepi, M., Verveld, M. J., Karcher, N. W., Franz, T., Abu-Zurayk, M., Görtz, S., and Kier, T. M., "Reduced-order models for aerodynamic applications, loads and MDO," *CEAS Aeronautical Journal*, Vol. 9, No. 1, 2018, pp. 171–193. doi:10.1007/s13272-018-0283-6.

[2] Sturdza, P., "Extensive supersonic natural laminar flow on the aerion business jet," *Collection of Technical Papers - 45th AIAA Aerospace Sciences Meeting*, Vol. 12, No. January, 2007, pp. 8465–8481. doi:10.2514/6.2007-685.

[3] Kwak, D., Ohira, K., Gaffuli, M., and Brezillon, J., "CFD validation for the supersonic transport configuration at low-speed and high alpha flight condition," *31st AIAA Applied Aerodynamics Conference*, 2013, pp. 1–11. doi:10.2514/6.2013-3030.

[4] Benner, P., and Willcox, K., *A Survey of Projection-Based Model Reduction Methods for Parametric Dynamical Systems*, Vol. 57, 2015.

[5] Brunton, S. L., Noack, B. R., and Koumoutsakos, P., "Machine learning for fluid mechanics," *Annual Review of Fluid Mechanics*, Vol. 52, 2020, pp. 477–508.

[6] Raissi, M., Perdikaris, P., and Karniadakis, G. E., "Physics-Informed Neural Networks: A Deep Learning Framework for Solving Forward and Inverse Problems Involving Nonlinear Partial Differential Equations," *Journal of Computational Physics*, 2019, pp. 1–18.

[7] Schmidt, M., and Lipson, H., "Distilling Free-form Natural Laws from experimental data," *Science*, Vol. 324, No. April, 2009, pp. 81–85.

[8] Rajaram, D., Puranik, T. G., Perron, C., and Mavris, D. N., "Non-intrusive parametric reduced order modeling using randomized algorithms," *AIAA Scitech 2020 Forum*, 2020, p. 0417.

[9] 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*, 2021, p. 3050. doi:10.2514/6.2021-3050.

[10] Bozeman Jr., M. D., "A Reduced Order Modeling Methodology for the Multidisciplinary Design and Analysis of Boundary Layer Ingestion Configurations," Ph.D. thesis, Georgia Institute of Technology, 2019.

[11] Behere, A., Rajaram, D., Puranik, T. G., Kirby, M., and Mavris, D. N., "Reduced Order Modeling Methods for Aviation Noise Estimation," *Sustainability*, Vol. 13, No. 3, 2021, p. 1120.

[12] Lumley, J. L., "The structure of inhomogeneous turbulent flows," *Atmospheric turbulence and radio wave propagation*, 1967.

[13] Fossati, M., "Evaluation of aerodynamic loads via reduced-order methodology," *AIAA Journal*, Vol. 53, No. 8, 2015, pp. 2389–2405. doi:10.2514/1.J053755.

[14] Rajaram, D., Gautier, R. H., Perron, C., Pinon-Fischer, O. J., and Mavris, D., "Non-Intrusive Parametric Reduced Order Models with High-Dimensional Inputs via Gradient-Free Active Subspace," *AIAA AVIATION 2020 FORUM*, 2020, p. 3184.

[15] Perron, C., Rajaram, D., and Mavris, D., "Development of a Multi-Fidelity Reduced-Order Model Based on Manifold Alignment," *AIAA AVIATION 2020 FORUM*, 2020, p. 3124.

[16] 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*, Vol. 477, No. 2253, 2021, p. 20210495.

[17] Rajaram, D., "Methods for Construction of Surrogates For Computationally Expensive High-Dimensional Problems," Ph.D. thesis, Georgia Institute of Technology, 2020.

[18] 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. doi:10.2514/1.J056060.

[19] Decker, K., Schwartz, H. D., and Mavris, D., "Dimensionality Reduction Techniques Applied to the Design of Hypersonic Aerial Systems," *AIAA AVIATION 2020 FORUM*, 2020, p. 3003.

[20] Franz, T., et al., "Interpolation-Based Reduced Order Modleing for Steady Transonic Flows via Manifold Learning," *International Journal of Computational Fluid Dynamics*, 2014.

[21] Herrmann, U., "Multiple Discipline Optimization and Aerodynamic Off-Design Analysis of Supersonic Transport Aircraft," *Journal of Aircraft*, 2008. doi:10.2514/1.32673.

[22] Ohlberger, M., and Rave, S., "Nonlinear reduced basis approximation of parameterized evolution equations via the method of freezing," *Comptes Rendus Mathematique*, Vol. 351, No. 23-24, 2013, p. 901–906. doi:10.1016/j.crma.2013.10.028, URL `http://dx.doi.org/10.1016/j.crma.2013.10.028`.

[23] Nair, N. J., and Balajewicz, M., "Transported snapshot model order reduction approach for parametric, steady-state fluid flows containing parameter-dependent shocks," *International Journal for Numerical Methods in Engineering*, Vol. 117, No. 12, 2019, pp. 1234–1262. doi:10.1002/nme.5998.

[24] Rim, D., and Mandli, K. T., "Displacement interpolation using monotone rearrangement," *arXiv*, Vol. 6, No. 4, 2017, pp. 1503–1531.

[25] Welper, G., "Interpolation of Functions with Parameter Dependent Jumps with Transformed Snapshots," *SIAM*, Vol. 48, No. 1, 2017, pp. 162–186.

[26] Franz, T., "Reduced-Order Modeling for Steady Transonic Flows via Manifold Learning," Ph.D. thesis, Deutches Zentrum fur Luft- und Raumfahrt, 2016.

[27] Lucia, D. J., Kingt, P. I., Reran, P. S., and Oxley, M. E., "Reduced order modeling for a one-dimensional nozzle flow with moving shocks," *15th AIAA Computational Fluid Dynamics Conference*, , No. June, 2001, pp. 1–11. doi:10.2514/6.2001-2602.

[28] Lucia, D. J., King, P. I., and Beran, P. S., "Domain decomposition for reduced-order modeling of a flow with moving shocks," *AIAA Journal*, Vol. 40, No. 11, 2002, pp. 2360–2362. doi:10.2514/2.1576.

[29] Buffoni, M., Telib, H., and Iollo, A., "Iterative methods for model reduction by domain decomposition," *Computers and Fluids*, Vol. 38, No. 6, 2009, pp. 1160–1167. doi:10.1016/j.compfluid.2008.11.008, URL `http://dx.doi.org/10.1016/j.compfluid.2008.11.008`.

[30] Baiges, J., Codina, R., and Idelsohn, S., "A domain decomposition strategy for reduced order models. Application to the incompressible Navier-Stokes equations," *Computer Methods in Applied Mechanics and Engineering*, Vol. 267, No. December, 2013, pp. 23–42. doi:10.1016/j.cma.2013.08.001.

[31] Sun, Z., Wang, C., Zheng, Y., and Bai, J., "Non-intrusive reduced-order model for predicting transonic flow with varying geometries," *Chinese Journal of Aeronautics*, Vol. 33, No. 2, 2020, pp. 508–519. doi:10.1016/j.cja.2019.12.014, URL `https://doi.org/10.1016/j.cja.2019.12.014`.

[32] Xiao, D., Fang, F., Heaney, C. E., Navon, I. M., and Pain, C. C., "A domain decomposition method for the non-intrusive reduced order modelling of fluid flow," *Computer Methods in Applied Mechanics and Engineering*, Vol. 354, 2019, pp. 307–330. doi:10.1016/j.cma.2019.05.039, URL `https://doi.org/10.1016/j.cma.2019.05.039`.

[33] Kerfriden, P., Goury, O., Rabczuk, T., and Bordas, S. P., "A partitioned model order reduction approach to rationalise computational expenses in nonlinear fracture mechanics," *Computer Methods in Applied Mechanics and Engineering*, Vol. 256, 2013, pp. 169–188. doi:10.1016/j.cma.2012.12.004, URL `http://dx.doi.org/10.1016/j.cma.2012.12.004`.

[34] Eldred, M., and Dunlavy, D., "Formulations for surrogate-based optimization with data fit, multifidelity, and reduced-order models," *11th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, 2006, p. 7117.

[35] Everson, R., and Sirovich, L., "Karhunen-Lòeve procedure for gappy data," *J. Opt. Soc. Am. A*, Vol. 12, No. 8, 1995, p. 1657.

[36] Bui-Thanh, T., Damodaran, M., Avenue, N., Willcox, K., and Computational, A., "Proper Orthogonal Decomposition Extensions For Parametric Applications in Transonic Aerodynamics," *AIAA paper*, Vol. 4213, 2003, pp. 1–11.

[37] Willcox, K., "Unsteady flow sensing and estimation via the gappy proper orthogonal decomposition," *2nd AIAA Flow Control Conference*, 2004. doi:10.2514/6.2004-2415.

[38] Brunton, S., and Kutz, N., *Data-Driven Science and Engineering*, Vol. 53, Cambridge University Press, 2019.

[39] Bishop, C. M., *Pattern Recognition and Machine Learning (Information Science and Statistics)*, Springer-Verlag, Berlin, Heidelberg, 2006.

[40] Tropea, C., Yarin, A. L., and Foss, J. F., *Springer Handbook of Experimental Fluid Mechanics*, Springer-Verlag Berlin Heidelberg, 2007.

[41] Kerschen, G., et al., "The Method of Proper Orthogonal Decompostion for Dynamical Characterization and Order Reduction of Mechanical Systems: An Overview," *Nonlinear Dynamics*, Vol. 41, No. 1-3, 2005, pp. 147–169.

[42] Sirovich, L., "Turbiulence and the Dynamics of Coherent Structures: Part I: Coherent Structures," *Quarterly of Applied Mathematics*, Vol. XLV, No. 3, 1987, pp. 561–571.

[43] Cayton, L., "Algorithms for Manifold Learning," *University of California at San Diego Technical Report*, Vol. 12, No. 1-17, 2005, p. 1.

[44] Bernstein, A. V., and Kuleshov, A. P., "Tangent bundle manifold learning via Grassmann & Stiefel eigenmaps," *arXiv preprint arXiv:1212.6031*, 2012.

[45] Tenenbaum, J. B., DeSilva, V., and Langford, J. C., "A Global Geometric Framework for Nonlinear Dimensionality Reduction," *Science*, Vol. 290, No. 5500, 2000, pp. 2319–2323.

[46] Maaten, L. v., Postma, E., and Herik, J. v., "Dimensionality Reduction: A Comparative Review," *Journal of Machine Learning Research*, Vol. 10, No. 66–71, 2009, pp. 1–35.

[47] Izenman, A. J., "Introduction to Manifold Learning," *Wiley Interdisciplinary Reviews: Computational Statistics*, 2012.

[48] Bui-Thanh, T., Damodaran, M., and Wilcox, K., "Proper Orthogonal Decomposition Extensions for Parametric Applications in Compressible Aerodynamics," *21st Applied Aerodynamics Conference*, 2003.

[49] Rajaram, D., Puranik, T. G., Ashwin Renganathan, S., Sung, W., Fischer, O. P., Mavris, D. N., and Ramamurthy, A., "Empirical assessment of deep gaussian process surrogate models for engineering problems," *Journal of Aircraft*, Vol. 58, No. 1, 2021, pp. 182–196.

[50] Fossati, M., "Evaluation of aerodynamic loads via reduced-order methodology," *AIAA Journal*, Vol. 53, No. 8, 2015, pp. 2389–2405.

[51] Mishne, G., Shaham, U., Cloninger, A., and Cohen, I., "Diffusion nets," *Applied and Computational Harmonic Analysis*, Vol. 47, No. 2, 2019, pp. 259–285.

[52] Park, K. H., Jun, S. O., Baek, S. M., Cho, M. H., Yee, K. J., and Lee, D. H., "Reduced-order model with an artificial neural network for aerostructural design optimization," *Journal of Aircraft*, Vol. 50, No. 4, 2013, pp. 1106–1116.

[53] Rasmussen, C. E., and Williams, C. K., *Gaussian Processes for Machine Learning*, Massachusetts Institute of Technology, 2006.

[54] Roweis, S. T., and Saul, L. K., "Nonlinear Dimensionality Reduction by Local Linear Embedding," *Science*, Vol. 290, No. 5500, 2000, pp. 2323–2326.

[55] Saul, L. K., and Roweis, S. T., "Think Globally, Fit Locally: Unsupervised Learning of Low-Dimensional Manifolds," *Journal of Machine Learning Research*, Vol. 4, 2003, pp. 119–155.

[56] Anderson Jr., J. D., *Fundamentals of Aerodynamics*, third edition ed., McGraw-Hill Higher Education, 2001.

[57] Sederberg, T. W., and Parry, S. R., "Free-form deformation of solid geometric models," *Proceedings of the 13th annual conference on Computer graphics and interactive techniques*, 1986, pp. 151–160.

[58] Palacios, F., Alonso, J., Duraisamy, K., Colonno, M., Hicken, J., Aranake, A., Campos, A., Copeland, S., Economon, T., Lonkar, A., Lukaczyk, T., and Taylor, T., "Stanford University Unstructured (SU^2): An open-source integrated computational environment for multi-physics simulation and design," 2013. doi:10.2514/6.2013-287, URL `https://doi.org/10.2514/6.2013-287`.

[59] Spalart, P., and Allmaras, S., "A one-equation turbulence model for aerodynamic flows," *30th aerospace sciences meeting and exhibit*, 1992, p. 439.

[60] Kolda, T. G., Lewis, R. M., and Torczon, V., "Optimization by direct search: New perspectives on some classical and modern methods," *SIAM Review*, Vol. 45, No. 3, 2003, pp. 385–482. doi:10.1137/S003614450242889.