Registration-based nonlinear model reduction of parametrized aerodynamics problems with applications to transonic Euler and RANS flows

Alireza H. Razavi[a,1,*], Masayuki Yano[a,2]

[a]*Institute for Aerospace Studies, University of Toronto, 4925 Duffein Street, Toronto, M3H 5T6, Ontario, Canada*

Abstract

We develop a registration-based nonlinear model-order reduction (MOR) method for partial differential equations (PDEs) with applications to transonic Euler and Reynolds-averaged Navier–Stokes (RANS) equations in aerodynamics. These PDEs exhibit discontinuous features, namely shocks, whose location depends on problem configuration parameters, and the associated parametric solution manifold exhibits a slowly decaying Kolmogorov N-width. As a result, conventional linear MOR methods, which use linear reduced approximation spaces, do not yield accurate low-dimensional approximations. We present a registration-based nonlinear MOR method to overcome this challenge. Our formulation builds on the following key ingredients: (i) a geometrically transformable parametrized PDE discretization; (ii) localized spline-based parametrized transformations which warp the domain to align discontinuities; (iii) an efficient dilation-based shock sensor and metric to compute optimal transformation parameters; (iv) hyperreduction and online-efficient output-based error estimates; and (v) simultaneous transformation and adaptive finite element training. Compared to existing methods in the literature, our formulation is efficiently scalable to larger problems and is equipped with error estimates and hyperreduction. We demonstrate the effectiveness of the method on two-dimensional inviscid and turbulent flows modeled by the Euler and RANS equations, respectively.

Keywords: discontinuous Galerkin method, adaptive high-order method, model reduction, hyperreduction, empirical quadrature procedure, aerodynamics

1. Introduction

We consider rapid and reliable solution of parametrized nonlinear partial differential equations (PDEs) with applications to transonic aerodynamic flows. In particular, we wish to rapidly evaluate quantities of interest (output), such as lift and drag, for any given flight condition parameter (input), such as the freestream Mach number; we also wish to provide reliable output predictions equipped with an error estimate. Our interest is in many-query scenarios, which require the input-output evaluation for many different parameter values and arise in engineering tasks including parameter sweep, optimization, and uncertainty quantification. We approach this many-query problem using projection-based model reduction based on offline-online computational decomposition: in the offline stage, we construct a reduced-order model (ROM) through a relatively expensive exploration of the parameter domain using a full-order model (FOM); in the online stage, we invoke the ROM to evaluate the solution field, output, and error estimate for any given input parameter value. This work focuses on model reduction for inviscid and turbulent transonic aerodynamic flows modeled respectively by the Euler and Reynolds-averaged Navier–Stokes (RANS) equations, which exhibit shocks whose location depends on the problem parameter.

*Corresponding author

Email addresses: a.razavi@mail.utoronto.ca (Alireza H. Razavi), masa.yano@utoronto.ca (Masayuki Yano)

[1]Graduate Student, Institute for Aerospace Studies, University of Toronto

[2]Associate Professor, Institute for Aerospace Studies, University of Toronto

We begin with a brief review of linear projection-based model reduction and its limitations.[3] We first introduce a d-dimensional spatial domain $\Omega \subset \mathbb{R}^d$, a P-dimensional parameter space $\mathcal{D} \subset \mathbb{R}^P$, a function space $\mathcal{V}$ over Ω, a parametrized solution $u : \mathcal{D} \to \mathcal{V}$, and the associated parametric solution manifold $\mathcal{U}_\mathcal{D} \coloneqq \{u(\mu)\}_{\mu\in\mathcal{D}} \subset \mathcal{V}$. We in addition introduce an N_h-dimensional FOM approximation space $\mathcal{V}_h \subset \mathcal{V}$ and the associated parametric solution manifold $\mathcal{U}_{\mathcal{D},h} \coloneqq \{u_h(\mu)\}_{\mu\in\mathcal{D}} \subset \mathcal{V}_h$. The standard *linear* projection-based model reduction methods (as described in e.g. [32, 9, 19]) collect FOM snapshots $\{u_h(\mu^{(i)})\}_{i=1}^{N_\text{train}}$ associated with training parameter set $\{\mu^{(i)}\}_{i=1}^{N_\text{train}}$, construct an $N \ll N_h$-dimensional reduced-basis (RB) $\{\phi_i\}_{i=1}^N$ using proper orthogonal decomposition (POD) or the Gram-Schmidt procedure, and find the ROM solution $u_N(\mu) \in \mathcal{V}_N \coloneqq \text{span}\{\phi_i\}_{i=1}^N$ through (Petrov-)Galerkin projection. Linear ROMs have the potential to produce, for all $\mu \in \mathcal{D}$, the solution $u_N(\mu)$ that converges rapidly to $u_h(\mu)$ *if* the parametric manifold $\mathcal{U}_{\mathcal{D},h}$ is amenable to low-dimensional approximation, in the sense that the Kolmogorov N-width of $\mathcal{U}_{\mathcal{D},h}$, defined as

$$\epsilon_N(\mathcal{U}_{\mathcal{D},h}) \coloneqq \inf_{\substack{\mathcal{V}_N\subset\mathcal{V}\\ \dim(\mathcal{V}_N)=N}} \sup_{u_h\in\mathcal{U}_{\mathcal{D},h}} \inf_{w_N\in\mathcal{V}_N} \|u_h - w_N\|_\mathcal{V},$$

decays rapidly with N. This assumption of the rapid decay of $\epsilon_N(\mathcal{U}_{\mathcal{D},h})$ holds for many problems, which has led to to the success of linear ROMs [9] including in aerodynamics [39]. However, if the solution $u(\mu)$ has a sharp feature, such as a shock, whose location depends on the parameter μ, $\epsilon_N(\mathcal{U}_{\mathcal{D},h})$ decays slowly and *linear* ROMs are ineffective [29]. This fundamental limitation of *linear* model reduction is referred to as the Kolmogrov barrier. The only way to overcome the Kolmogorov barrier is to consider nonlinear ROMs (NLROMs), which use *nonlinear* approximations.

Many different approaches to construct nonlinear approximation spaces have recently been proposed, including those based on piecewise linear approximations (e.g., [14, 2]), piecewise quadratic approximation [4], and more general nonlinear approximations using neural networks of various forms (e.g., [24, 16, 23, 5]). In this work, we focus on registration-based or Lagrangian methods (e.g., [28, 20, 37]). While specific formulation and implementation, as well as their interpretation, vary among the works, registration-based methods build on several common ingredients: a diffeomorphism $g(\cdot;\nu) : \Omega \to \Omega$, which is parametrized by a transformation parameter $\nu \in \mathcal{D}_\nu$ and warps points in the domain Ω while preserving the boundary $\partial\Omega$; a function $\xi : \mu \mapsto \nu$ that identifies an appropriate transformation parameter $\nu \in \mathcal{D}_\nu$ for each problem parameter $\mu \in \mathcal{D}$; and a metric that measures the distance between a reference solution $u(\mu^\text{ref})$ and transformed solution $u(\mu) \circ g(\cdot;\nu \equiv \xi(\mu))$ to identify an optimal ξ. The main idea of registration-based methods is to transform the solution $u(\mu)$ for each $\mu \in \mathcal{D}$ using μ-parametrized transformation $g(\cdot;\nu \equiv \xi(\mu))$ so that sharp features in $u(\mu)$ that are not amenable to linear compression are aligned with sharp features in $u(\mu^\text{ref})$ for the transformed solution $u(\mu)\circ g(\cdot;\xi(\mu))$. In other words, we choose $g(\cdot;\nu \equiv \xi(\mu))$ such that the transformed parametric solution manifold $\{u(\mu) \circ g(\cdot;\xi(\mu))\}_{\mu\in\mathcal{D}}$ is more amenable to linear approximation than the original manifold $\{u(\mu)\}_{\mu\in\mathcal{D}}$. The effectiveness of the approach for problems that exhibit parameter-dependent sharp features and discontinuities has been shown in theory and in practice [28, 20, 37, 10, 34, 21]. More recently, registration-based methods have also been successfully applied to projection-based model reduction of inviscid transonic flows modeled by the Euler equations [27, 36, 15, 25, 12]. However, to the best of our knowledge, registration-based methods have not been demonstrated for turbulent transonic aerodynamics problems of industrial relevance, and there remains a number of challenges to achieve this goal.

The first challenge, and our first contribution, is the development of a scalable, parallelizable, and robust approach to find a parametrized transformation $g(\cdot;\xi(\mu)) : \Omega \to \Omega$ for transonic turbulent flows. Most registration-based methods use the $L^2(\Omega)$ norm of the differences in the full state fields $u(\mu)$ [28, 20, 37, 10, 34, 27] or the Mach number fields $M(u(\mu))$ [36, 15] as the metric to minimize to align sharp features: i.e., $\min_\nu \|u(\mu^\text{ref}) - u(\mu) \circ g(\cdot;\nu)\|_{L^2(\Omega)}$ or $\min_\nu \|M(u(\mu^\text{ref})) - M(u(\mu)) \circ g(\cdot;\nu)\|_{L^2(\Omega)}$. However, this approach suffers from two issues. First, the sensor targets the most dominant features in the solution even if some of those features are amenable to linear compression, and, conversely, it might not align less dominant features that benefit from alignment. While this is not a significant issue for scalar or Euler equations where the

[3] "Linear" refers to the fact that the approximation space is linear; the governing PDE may be nonlinear.

discontinuity/shock is the dominant feature, this is a problem for turbulent flows modeled by the RANS equations, which exhibit many different features. Second, whether the solution is explicitly transformed through Lagrangian mapping (e.g., [27]) or implicitly through mesh transformation (e.g., [36]), the evaluation of the $L^2(\Omega)$ norm of the difference between reference and transformed fields on distributed-memory parallel computers requires complex operations with significant communication across subdomain boundaries. This can be prohibitive in larger scale, industrially relevant problems. We address these two issues using a combination of (i) a dilation-based shock sensor [26], which isolates the shock from other dominant features to focus our registration process, (ii) a localized transformation, which isolates the regions with shocks and still maintains global C^1 continuity, and (iii) a shock feature indicator based on principal points, which eliminates the need to solve the parallel L^2-norm minimization problem. (We note that [25] overcome the issues using a specialized FOM that natively performs shock alignment and [12] overcomes the issue by leveraging a specific mesh topology, but these methods require a specialized FOM or problem setup.)

Our second contribution is the development and assessment of a registration-based method that (i) provides online-efficient a posteriori error estimation for the quantity of interest and (ii) provides significant online computational speedup through a hyperreduction method tailored for high-order methods. To achieve (i), we appeal to the dual-weighted residual (DWR) method [8] and specifically it adopted to ROMs [38]. To achieve (ii), we appeal to the empirical quadrature procedure (EQP) [41], and in particular its point-wise variant specialized for high-order discontinuous Galerkin (DG) methods [13]. To incorporate these ingredients developed for linear ROMs, we reinterpret nonlinear model reduction based on solution transformation as a linear model reduction applied to transformed PDEs, which enables straightforward formulation and implementation of the techniques.

Our third contribution is the demonstration of the proposed registration-based model reduction approach on inviscid *and* turbulent transonic flows modeled by the Euler *and* RANS equations. As noted earlier, to the best of our knowledge, the registration-based method has not been applied to RANS problems. We demonstrate that nonlinear approximation significantly reduces the dimension of the reduced space required to achieve a given error level, point-wise hyperreduction provides significant online speedup, and the output error estimate is effective. We also demonstrate that, unlike neural network based methods, our method does not require extensive offline training to achieve an accurate online approximation.

The remainder of the work is organized as follows. Section 2 presents the registration-based nonlinear approximation formulation and introduces its key ingredients: the transformed problem, parametrized geometry transformation, shock-location indicator, and transformation optimization. Section 3 presents discretizations of the transformed problem: a FOM based on an adaptive high-order DG method; and a hyperreduced ROM based on POD, point-wise EQP, and DWR-based error estimate. Section 4 assesses the accuracy and cost of the proposed formulation using inviscid and turbulent transonic flows modeled by the Euler and RANS equations.

2. Nonlinear approximation and problem transformation

2.1. Problem statement

We first introduce the problem that we consider throughout this work. We introduce a P-dimensional parameter domain $\mathcal{D} \subset \mathbb{R}^P$, and a d-dimensional spatial domain $\Omega \subset \mathbb{R}^d$. The parametrized system of m PDEs and boundary conditions that we consider is of the form

$$\begin{aligned} \nabla \cdot F(\hat{u}(\mu);\mu) - \nabla \cdot (K(\hat{u}(\mu);\mu)\nabla\hat{u}(\mu)) &= S(\hat{u}(\mu), \nabla\hat{u}(\mu);\mu) \quad \text{in } \Omega, \\ B(\hat{u}(\mu), \hat{n} \cdot K(\hat{u}(\mu);\mu)\nabla\hat{u}(\mu);\mu) &= 0 \quad \text{in } \partial\Omega, \end{aligned} \tag{1}$$

where $\hat{u} : \Omega \to \mathbb{R}^m$ is the solution that we seek[4], $F : \mathbb{R}^m \times \mathcal{D} \to \mathbb{R}^{m\times d}$ is the convective flux, $K : \mathbb{R}^m \times \mathcal{D} \to \mathbb{R}^{m\times d\times m\times d}$ is the diffusion tensor, $S : \mathbb{R}^m \times \mathbb{R}^{m\times d} \times \mathcal{D} \to \mathbb{R}^m$ is the source function, $B : \mathbb{R}^m \times \mathbb{R}^{m\times d} \times \mathbb{R}^d \times \mathcal{D} \to$

[4]We denote the solution with $\hat{\cdot}$ so that we can reserve the notation without the $\hat{\cdot}$ for the transformed problem.

$\mathbb{R}^m$ is the boundary operator, and $\hat{n}$ is the outward-pointing normal vector on $\partial\Omega$. Given the solution, we evaluate the functional output

$$s(\mu) \coloneqq \widehat{J}(\hat{u}(\mu); \mu),$$

where $\widehat{J}(\cdot; \mu)$ is the parametrized output functional. In this work, we focus on transonic flows governed by the Euler and RANS equations with the aerodynamic lift or drag as the functional output.

2.2. Transformed problem

We now introduce a transformed problem that we obtain by warping the spatial domain Ω. By way of preliminaries, we introduce a parametrized diffeomorphism $g(\cdot; \nu) : \Omega \to \Omega$, where $\nu \in \mathcal{D}_\nu$ is the parameter of the transformation, and $\mathcal{D}_\nu \subset \mathbb{R}^{N_\nu}$ is the domain of transformation parameters; we will specify the specific form of g in Section 2.3. The system of PDEs on the transformed domain is given by

$$\begin{aligned} \nabla_\nu \cdot F(u(\mu, \nu); \mu) - \nabla_\nu \cdot (K(u(\mu, \nu); \mu) \nabla_\nu u(\mu, \nu)) &= S(u(\mu, \nu), \nabla_\nu u(\mu, \nu); \mu) \quad \text{in } \Omega, \\ B(u(\mu; \nu), n_\nu \cdot K(u(\mu, \nu); \mu) \nabla_\nu u(\mu; \nu); \mu) &= 0 \quad \text{in } \partial\Omega, \end{aligned} \tag{2}$$

where $\nabla_\nu \coloneqq (\partial_{\nu,x_1}, \ldots, \partial_{\nu,x_d})$ is the transformed gradient that satisfies $\partial_{\nu,x_j} = \sum_{i=1}^d (Dg^{-1}(\cdot; \nu))_{ij} \partial_{x_i}$, $n_\nu \coloneqq Dg(\cdot; \nu)^{-T} \hat{n} / \|Dg(\cdot; \nu)^{-T} \hat{n}\|_2$ is the transformed outward-pointing normal vector, and $Dg(\cdot; \nu) : \Omega \to \mathbb{R}^{d \times d}$ is the Jacobian of the transformation associated with $g(\cdot; \nu) : \Omega \to \Omega$.

The solution $u(\mu; \nu)$ to the transformed problem (2) and the solution $\hat{u}(\mu)$ to the original problem (1) are related by $u(\mu; \nu) = \hat{u}(\mu) \circ g(\cdot; \nu)$ for all $\mu \in \mathcal{D}$ and $\nu \in \mathcal{D}_\nu$. Hence, given the solution $u(\mu, \nu)$ to the transformed problem, we may readily find the solution $\hat{u}(\mu)$ to the original problem through the inverse transformation. Similarly, we equip the transformed problem with the transformed output functional $J(\cdot; \mu, \nu)$ so that $s(\mu) = \widehat{J}(\hat{u}(\mu); \mu) = J(u(\mu, \nu); \mu, \nu)$ for all $\mu \in \mathcal{D}$ and $\nu \in \mathcal{D}_\nu$.

As discussed in the introduction, our goal is to find transformations which make the transformed problems more amenable to RB approximation than the original problem. To formalize the idea, we first introduce the parametric solution manifold for the original problem:

$$\widehat{\mathcal{U}}_\mathcal{D} = \{\hat{u}(\mu)\}_{\mu \in \mathcal{D}}.$$

We next introduce the parametric solution manifold for the transformed problem:

$$\mathcal{U}_{\mathcal{D},\xi} = \{u(\mu, \xi(\mu)) \coloneqq \hat{u}(\mu) \circ g(\cdot; \xi(\mu))\}_{\mu \in \mathcal{D}},$$

where $\xi : \mathcal{D} \to \mathcal{D}_\nu$ is a function that maps the (problem) parameter $\mu \in \mathcal{D}$ to the transformation parameter $\nu \in \mathcal{D}_\nu$. We wish to choose the μ-parametrized transformation $g(\cdot; \xi(\mu))$ — which is defined by a ν-parametrized transformation $g(\cdot; \nu)$ and an associated parameter map $\mathcal{D} \ni \mu \mapsto \xi(\mu) \equiv \nu \in \mathcal{D}_\nu$ — so that the Kolmogorov N-width of $\mathcal{U}_{\mathcal{D},\xi}$ given by

$$\epsilon_N(\mathcal{U}_{\mathcal{D},\xi}) \coloneqq \inf_{\substack{\mathcal{V}_N \subset \mathcal{V} \\ \dim(\mathcal{V}_N) = N}} \sup_{u \in \mathcal{U}_{\mathcal{D},\xi}} \inf_{w_N \in \mathcal{V}_N} \|u - w_N\|_\mathcal{V}$$

decays (much) more rapidly than that of $\widehat{\mathcal{U}}_\mathcal{D}$.

Remark 1. Whether there exists a transformation $g(\cdot; \xi(\mu))$ that results in a much more rapid decay of the Kolmogorov N-width is problem dependent, and understanding for which classes of problems the transformation is effective is an important question on its own. However, we do not attempt to answer this general question in this work. We instead focus on transonic flow problems with parameter-dependent shocks, which are known to benefit from the transformation.

2.3. Parametrized geometry transformation

We now introduce a family of parametrized transformations $g(\cdot;\nu) : \Omega \to \Omega$ that we consider in this work. Our choice of transformation is motivated by the fact that we focus on problems where discontinuities are localized to a subregion of the domain Ω. For instance, in a typical transonic flow over an airfoil, the shock is localized to a small region over the suction side of the airfoil. To this end, we introduce the following space of admissible transformations.

Definition 2 (admissible transformations)**.** The geometry transformation that we consider in this work is given by

$$\mathcal{G} \coloneqq \{g : \Omega \to \Omega \mid g|_\omega = g_\omega \in \mathcal{G}_\omega,\ g|_{\Omega\setminus\omega} = I_d\}$$

where $\omega \subset \Omega$ is a subdomain that is warped, I_d is the identity map, and $\mathcal{G}_\omega$ is the space of local transformations such that $g_\omega \in \mathcal{G}_\omega$ satisfies the following conditions:

C1. $\det(\nabla g_\omega) > 0$

C2. $g_\omega \in C^1(\omega)$

C3. $g_\omega(s) = s, \quad s \in \partial\omega \setminus \partial\Omega$

C4. $\nabla g_\omega(s) = I_d, \quad s \in \partial\omega \setminus \partial\Omega$

C5. $g_\omega(s) \in \partial\omega \cap \partial\Omega, \quad s \in \partial\omega \cap \partial\Omega$

We make a few observations. First, the "global map" $g \in \mathcal{G}$ is defined in a piecewise manner, with the (non-identity) transformation applied only to the subdomain $\omega \subset \Omega$. Second, C5 allows the points on $\partial\omega \subset \partial\Omega$ (i.e., the boundary of the subdomain ω that is on the domain boundary $\partial\Omega$) to slide along the boundary, which is important when the discontinuity abuts $\partial\Omega$. Third, the five conditions on the local transformation ensure that the global map is a $C^1(\Omega)$-diffeomorphism, as the following proposition shows:

Proposition 3. *A map $g \in \mathcal{G}$ defined by Definition 2 is a $C^1(\Omega)$-diffeomorphism that maps Ω to itself.*

Proof. We first observe that the local transformation $g_\omega \in \mathcal{G}_\omega$ is a C^1-diffeomorphism on ω since C1 and C2 ensure that g_ω is invertible and C^1 continuous, respectively. We next observe that the transformation $g \in \mathcal{G}$ is a C^1-diffeomorphism on $\Omega\setminus\omega$ since it is the identity map. We also observe that C3 and C4 ensure that the value and first derivative of the map is continuous across the interface $\partial\omega\setminus\partial\Omega$. The combination of these three observations ensures that $g \in \mathcal{G}$ is a $C^1(\Omega)$-diffeomorphism. Finally, C5 ensures that $g_\omega(\partial\omega \setminus \partial\Omega) = \partial\omega \setminus \partial\Omega$, which, in conjunction with C3, ensure that g_ω maps ω to itself. □

Remark 4. The fact that $g \in \mathcal{G}$ is a $C^1(\Omega)$-diffeomorphism ensures that, if the original solution $u(\mu)$ is C^1 continuous, then the transformed solution $u(\mu;\nu) \equiv \hat{u}(\mu) \circ g(\cdot;\nu)$ is also C^1 continuous. In other words, $g \in \mathcal{G}$ does not introduce kinks in the solution, particularly across the subdomain interface $\partial\omega$. Such kinks can negatively affect the numerical approximation of the transformed PDE, when the standard "reference domain" or "map-then-discretize" approach is used; see, e.g., [35, 32].

We now introduce a specific class of maps that we consider in this work. To construct a parametrized transformation that belongs to the space $\mathcal{G}_\omega$, we consider a composition of two maps: (i) transfinite interpolation, which allows us to map a desired subregion ω (with potentially complicated geometry) to a relatively simple unit domain over which another transformation is applied; and (ii) spline-based mapping, which induces parametrized transformation on the unit domain that satisfies the required regularity conditions of Definition 2. The overview of the composite map constructed from these two transformations is illustrated in Figure 1. The use of the composition follows from [15, 21], but we consider different specific maps to meet the regularity conditions. There are two advantages to this construction: we can apply geometry transformation to more complex geometry in a straightforward manner by transforming the geometry to the unit domain using (i); and we can construct (ii) on a regular uniform grid rather than an arbitrary grid which conforms to the complex geometry.

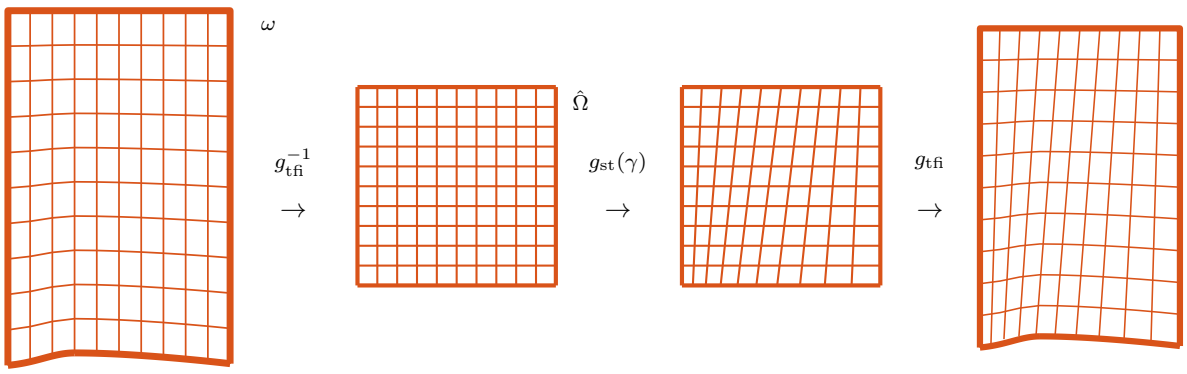


Figure 1: Illustration of the transformation $g_\omega(\cdot;\nu) \equiv g_{\text{tfi}} \circ g_{\text{st}}(\cdot;\nu) \circ g_{\text{tfi}}^{-1}$ constructed from g_{tfi} and g_{st}.

We first introduce a transfinite interpolation $g_{\text{tfi}} : [0,1]^d \to \omega$, which maps the unit d-cube to ω using linear (resp. bilinear and trilinear) functions in one (resp. two and three) dimension. For instance, in two dimensions, the transfinite interpolation $g_{\text{tfi}} : [0,1]^{d\equiv 2} \to \omega \subset \mathbb{R}^2$ is given by

$$\begin{aligned} g_{\text{tfi}}(s) = (1-s_1)b_{\text{L}}(s_2) + s_1 b_{\text{R}}(s_2) + (1-s_2)b_{\text{B}}(s_1) + s_2 b_{\text{T}}(s_1) \\ - (1-s_1)(1-s_2)b_{\text{BL}} - (1-s_1)s_2 b_{\text{TL}} - s_1(1-s_2)b_{\text{BR}} - s_1 s_2 b_{\text{TR}}, \end{aligned}$$

where $b_\square : \mathbb{R}^1 \to \mathbb{R}^2$ for $\square \in \{\text{L}, \text{R}, \text{B}, \text{T}\}$ provides the parametrized coordinates along the left (L), right (R), bottom (B), and top (T) boundaries, and $b_\circ \in \mathbb{R}^2$ for $\circ \in \{\text{BL}, \text{TL}, \text{BR}, \text{TR}\}$ provides the coordinates of the four corner points. We note that the sole purpose of this transformation is to map the d-cube to ω, and hence this transformation is not parametrized. In addition, we *assume* that the boundary $\partial\omega$ is sufficiently close to $\partial[0,1]^d$, where $\partial[0,1]^d$ denotes the boundary of the d-cube, so that the transformation g_{tfi} satisfies $\det(\nabla g_{\text{tfi}}) > 0$ and is invertible. This assumption is typically valid when the number of corners in ω match the number of corners in the d-cube and when the corners of ω are connected by C^1 continuous edges.

We next introduce a ν-parametrized spline-based transformation $g_{\text{st}}(\cdot;\nu) : [0,1]^d \to [0,1]^d$. Specifically, we introduce an equi-spaced Cartesian grid over the d-cube and the associated (cardinal and C^2-continuous) cubic B-spline over the grid. In one dimension, each cubic spline basis function has its support over the four segments delineated by five grid nodes, where the control point is the grid node at the center. In two and three dimensions, we obtain a spline basis function through the tensor product of one-dimensional spline basis functions with support over 4^d grid elements delineated by 5^d grid nodes. To construct a parametrized transformation, we associate the parameter $\nu \in \mathcal{D}_\nu \subset \mathbb{R}^{N_\nu}$ with the value of the B-spline at a set of active control points and in active control directions. In other words,

$$g_{\text{st}}(s;\nu) = s + \sum_{j=1}^{N_\nu} \nu_j B_j(s),$$

where $B_j : [0,1]^d \to \mathbb{R}^d$ is a (vector-valued) spline basis function associated with an active control point and direction. To ensure that the composite map $g_\omega(\cdot;\nu)$ that uses $g_{\text{st}}(\cdot;\nu)$ ultimately satisfies the desired conditions in Definition 2, we choose the nodes on B_{int} and their immediate neighbors as inactive nodes, where $B_{\text{int}} \coloneqq \partial[0,1]^d \setminus g_{\text{tfi}}^{-1}(\partial\Omega)$ is the subset of $\partial[0,1]^d$ that interfaces with the rest of the domain $\Omega \setminus \omega$ after the transfinite interpolation. This ensures $\sum_{j=1}^{N_\nu} \nu_j B_j(s) = 0$ for $s \in B_{\text{int}}$, which in turn implies $g_{\text{st}}(s;\nu) = s$. In addition, since B_j has vanishing first and second derivatives at the edge of its support, $\nabla g_{\text{st}}(s;\nu) = I_d$ and $\nabla^2 g_{\text{st}}(s;\nu) = 0$ for $s \in B_{\text{int}}$. In addition, we *assume* that the parameter domain $\mathcal{D}_\nu$ is chosen such that $\nu \in \mathcal{D}_\nu$ is sufficiently small and $g_{\text{st}}(\cdot;\nu)$ satisfies $\det(\nabla g_{\text{st}}(\cdot;\nu)) > 0$ and is invertible. In practice, this can also be achieved by choosing sufficiently large ω so that shocks will not be close to the boundary $g_{\text{tfi}}^{-1}(\partial\Omega)$ for all $\mu \in \mathcal{D}$.

Remark 5. The construction of $g_{\text{st}}(\cdot;\nu)$ highlights the need for g_{tfi}. The typical construction of $g_{\text{st}}(\cdot;\nu)$ uses a grid of uniformly distributed nodes in $[0,1]^d$. While it may be straightforward to affinely scale this transformation to arbitrarily sized cubes, it would require much more careful construction for arbitrary geometry. On the other hand, with g_{tfi}, the arbitrary geometry is warped to $[0,1]^d$.

Remark 6. We choose the active control points and directions to match the shape and direction of the motion of shocks. For example, in a problem where we expect predominantly stream-wise variation of the shock location, we can choose to only activate those control points and directions that correspond to this motion. Restricting the candidate motions based on prior knowledge can improve the accuracy of alignment by limiting unnecessary warping.

We may now state the ν-parametrized local transformation $g_\omega \in \mathcal{G}_\omega$ as a composition:

$$g_\omega(\cdot;\nu) \equiv g_{\text{tfi}} \circ g_{\text{st}}(\cdot;\nu) \circ g_{\text{tfi}}^{-1}.$$

In words, g_{tfi}^{-1} maps ω to the d-cube, then $g_{\text{st}}(\cdot;\nu)$ affects a ν-parametrized transformation, and finally g_{tfi} maps the d-cube with internal transformation back to ω. We again refer to Figure 1 for an illustration of the composite map. The following proposition shows that the composition yields a local transformation with desired properties:

Proposition 7. *Assuming g_{tfi} and $g_{\text{st}}(\cdot;\nu)$ are invertible, the transformation $g_\omega(\cdot;\nu) \equiv g_{\text{tfi}} \circ g_{\text{st}}(\cdot;\nu) \circ g_{\text{tfi}}^{-1}$ is in $\mathcal{G}_\omega$ defined in Definition 2.*

Proof. The condition C1 follows from the assumption that g_{tfi} and g_{st} are invertible. In addition, the condition C2 is satisfied because $g_{\text{tfi}} \in C^\infty((0,1)^d)$ and $g_{\text{st}}(\cdot;\nu) \in C^2(\omega)$, and hence $g_\omega \in C^1(\omega)$. Next, the condition C3 is satisfied because $g_{\text{st}}(s';\nu) = s'$ for $s' \in B_{\text{int}}$ and hence $g_\omega(s) = g_{\text{tfi}}(g_{\text{st}}(g_{\text{tfi}}^{-1}(s);\nu)) = g_{\text{tfi}}(g_{\text{tfi}}^{-1}(s)) = s$ for $s \in \partial\omega \setminus \partial\Omega$. Also, the condition C4 is satisfied because $g_{\text{st}}(s';\nu) = I_d$ for $s' \in B_{\text{int}}$ and hence $\nabla g_\omega(s) = \nabla g_{\text{tfi}}(s')\nabla g_{\text{st}}(s';\nu)\nabla g_{\text{tfi}}^{-1}(s) = \nabla g_{\text{tfi}}(s')\nabla g_{\text{tfi}}^{-1}(s) = I_d$ for $s \in \partial\omega \setminus \partial\Omega$ and $s' = g_{\text{tfi}}^{-1}(s)$. Finally, the condition C5 is satisfied because $g_{\text{st}}(s';\nu) \in B_{\text{bnd}}$ for $s' \in B_{\text{bnd}}$, where $B_{\text{bnd}} \coloneqq \partial[0,1]^d \cap g_{\text{tfi}}^{-1}(\partial\Omega)$, and hence $g_\omega(s) = g_{\text{tfi}}(g_{\text{st}}(g_{\text{tfi}}^{-1}(s))) \in \partial\omega \cap \partial\Omega$ for $s \in \partial\omega \cap \partial\Omega$.

□

2.4. Shock-location indicator

To find an appropriate transformation parameter $\nu \in \mathcal{D}_\nu$ that aligns discontinuities, we first need to systematically characterize the location of the discontinuities. In this work, we use a combination of (i) a shock indicator, which flags regions with shocks; (ii) a smoother, which regularizes and normalizes the (rough) raw output of the shock indicator; and (iii) statistical moments, which reduces the information in the regularized shock indicator to several key values.

We first introduce a dilation-based shock indicator [26] given by

$$s_{\text{dil}}(u) = -L\nabla_\nu \cdot v(u)/c^\star(u), \tag{3}$$

where $v(u) : \Omega \to \mathbb{R}^d$ is the velocity field associated with the state $u \in \mathcal{V}$, $c^\star(u)$ is the critical sound speed given by $c^\star(u) = \sqrt{2\gamma R T_0(u)/(\gamma+1)}$, γ is the ratio of specific heats, R is the gas constant, $T_0(u)$ is the stagnation temperature, and $L = kh/p$ is an empirically turned length scale for h the element size and p the polynomial degree. We note that $s_{\text{dil}}(u) : \Omega \to \mathbb{R}$ is a scalar-valued field that takes on a large positive values in the regions where the flow experiences strong compression, such as in a shock.

We next regularize the (raw) shock indicator field $s_{\text{dil}}(u) : \Omega \to \mathbb{R}$ to obtain a smoothed field $s_{\text{dil}}^\circ(u) : \Omega \to \mathbb{R}$. To this end, we solve the reaction-diffusion equation with homogeneous Neumann boundary conditions given by

$$\begin{aligned} -\alpha\Delta s_{\text{dil}}^\circ(u) + s_{\text{dil}}^\circ(u) &= (s_{\text{dil}}(u))^+ \quad \text{in } \Omega, \\ \hat{n} \cdot \nabla s_{\text{dil}}^\circ(u) &= 0 \quad \text{on } \partial\Omega, \end{aligned}$$

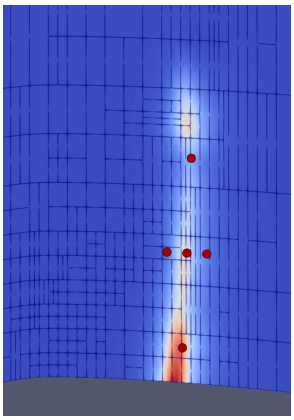


Figure 2: Example of sensor $\hat{s}^{\circ}_{\text{dil}}(u)$ and principal points $\{p_i(u)\}_{i=1}^{n_p}$ used to approximate the location of a shock.

where $\alpha \in \mathbb{R}_{>0}$ is a parameter that specifies the characteristic length scale for smoothing, and $(s_{\text{dil}}(u))^+ \coloneqq \max\{s_{\text{dil}}(u), 0\}$, so that the right-hand side is always positive, accounting for compression but not for rarefaction. This ensures that $s^{\circ}_{\text{dil}}(u) \geq 0$ everywhere. We then normalize the regularized shock indicator field $s^{\circ}_{\text{dil}}(u) : \Omega \to \mathbb{R}$. Namely, we set

$$\hat{s}^{\circ}_{\text{dil}}(u) \coloneqq \frac{s^{\circ}_{\text{dil}}(u)}{\int_{\Omega} s^{\circ}_{\text{dil}}(u) dx}.$$

The field $\hat{s}^{\circ}_{\text{dil}}(u) : \Omega \to \mathbb{R}$ is normalized and non-negative, which allows us to interpret $\hat{s}^{\circ}_{\text{dil}}(u)$ as a probability density on Ω.

We finally reduce the information contained in the shock indicator field $\hat{s}^{\circ}_{\text{dil}}(u) : \Omega \to \mathbb{R}$ to a set of principal points. To this end, we first compute the first and second central moments of the shock indicator field: $m(u) \in \mathbb{R}^d$ such that $m_i(u) \coloneqq \int_{\Omega} x_i \hat{s}^{\circ}_{\text{dil}}(u) dx$, and $\Sigma(u) \in \mathbb{R}^{d \times d}$ such that $\Sigma_{ij}(u) \coloneqq \int_{\Omega} (x_i - m_i(u))(x_j - m_j(u)) \hat{s}^{\circ}_{\text{dil}}(u) dx$ for $i, j = 1, \ldots, d$. We then compute the set of $n_p \coloneqq 2d+1$ principal points $\{p_k(u) \in \mathbb{R}^d\}_{k=1}^{n_p}$ as per the following definition.

Definition 8 (Principal points $\{p_k(u) \in \mathbb{R}^d\}_{k=1}^{n_p}$ of the sensor $\hat{s}^{\circ}_{\text{dil}}(u)$)**.** Given the first central moment $m(u) \in \mathbb{R}^d$ and second central moment $\Sigma(u) \in \mathbb{R}^{d \times d}$, the principal points $\{p_k(u) \in \mathbb{R}^d\}_{k=1}^{n_p}$ are given by

$$\{p_k(u)\}_{k=1}^{n_p} = \{m\} \cup \{m + \lambda_l^{1/2} \chi_l, m - \lambda_l^{1/2} \chi_l\}_{l=1}^{d}, \tag{4}$$

where $(\chi_l, \lambda_l) \in \mathbb{R}^d \times \mathbb{R}$, $l = 1, \ldots d$, are the eigenvectors and eigenvalues of $\Sigma(u)$.

The set of $2d + 1$ principal points captures the mean (i.e., center) of the shock, as well as the variance (i.e., stretching and orientation) of the shock. An illustrative example of the shock sensor and principal points in two dimensions is shown in Figure 2.

In summary, through the sequence of mappings—$\mu \mapsto u(\mu) \mapsto s_{\text{dil}}(u(\mu)) \mapsto \hat{s}^{\circ}_{\text{dil}}(u(\mu)) \mapsto \{p_k(u(\mu))\}_{k=1}^{n_p}$—, we have constructed a map from the parameter $\mu \in \mathcal{D}$ to the principal points $\{p_k(u(\mu)) \in \mathbb{R}^d\}_{k=1}^{n_p}$ for the shock indicator $\hat{s}^{\circ}_{\text{dil}}(u(\mu))$, which contains significantly reduced but essential information about the shock location, stretching, and orientation. That is, we approximate a shock by its principal points. As we will see, this approximation makes our method scalable to larger scale problems and in parallel settings.

2.5. Construction of optimal transformation parameter map: $\xi : \mathcal{D} \ni \mu \mapsto \nu \in \mathcal{D}_\nu$

We finally present a procedure to construct the map $\xi : \mathcal{D} \to \mathcal{D}_\nu$ from the problem parameter $\mu \in \mathcal{D}$ to the optimal transformation parameter $\nu \in \mathcal{D}_\nu$ that best aligns the shock in the solution $\hat{u}(\mu)$ to the shock in some reference solution $\hat{u}(\mu_{\text{ref}})$; i.e., $g^{-1}(p(\hat{u}(\mu)); \nu) \approx p(\hat{u}(\mu_{\text{ref}}))$. We learn this map using a set of training data. Namely, we first introduce a training parameter set $\mathcal{D}_{\text{train}} \coloneqq \{\mu^{(i)}\}_{i=1}^{N_{\text{train}}} \subset \mathcal{D}$; we choose the reference parameter μ_{ref} as $\mu \in \mathcal{D}_{\text{train}}$ that is closest to the centroid of $\mathcal{D}_{\text{train}}$. We next compute the associated

(untransformed) solutions $\{\hat{u}(\mu^{(i)})\}_{i=1}^{N_{\text{train}}}$. We then invoke the sequence of maps described in Section 2.4 to obtain the associated set of principal points $\{\{p_j(\hat{u}(\mu^{(i)}))\}_{j=1}^{n_p}\}_{i=1}^{N_{\text{train}}}$. We then solve the optimization problem: for $i = 1, \ldots, N_{\text{train}}$, find

$$\nu^\star(\mu^{(i)}) = \arg\min_{\nu \in \mathcal{D}_\nu} \sum_{j=1}^{n_p} \|p_j(\hat{u}(\mu_{\text{ref}})) - g^{-1}(p_j(\hat{u}(\mu^{(i)})); \nu)\|^2_{\ell^2(\mathbb{R}^d)}.$$

This provides a set of transformation parameters $\{\nu^\star(\mu^{(i)})\}_{i=1}^{N_{\text{train}}}$ associated with $\{\hat{u}(\mu^{(i)})\}_{i=1}^{N_{\text{train}}}$. To approximate the map $\xi : \mu \mapsto \nu$, we assume that $\mu \mapsto \nu$ is sufficiently regular and approximate the map using radial basis functions (RBFs) applied to each component of the training data $\{\mu \in \mathcal{D} \subset \mathbb{R}^P\}_{\mu \in \mathcal{D}_{\text{train}}} \mapsto \{\nu^\star(\mu) \in \mathcal{D}_\nu \subset \mathbb{R}^{N_\nu}\}_{\mu \in \mathcal{D}_{\text{train}}}$: i.e.,

$$\xi(\mu)_i = \text{RBF}(\mu; \{\nu^\star(\mu)_i\}_{\mu \in \mathcal{D}_{\text{train}}}), \quad i = 1, \ldots, N_\nu. \tag{5}$$

In practice, we use a variant of the above formulation because in our NLROM construction we do not approximate the solution $\hat{u}(\mu)$ to (1) but rather the solution $u(\mu, \nu)$ to the transformed problem (2). To this end, we first accompany the training parameter set $\mathcal{D}_{\text{train}}$ with an associated (in general non-optimal) training transformation parameter set $\mathcal{D}_{\nu,\text{train}} \coloneqq \{\nu^{(i)}\}_{i=1}^{N_{\text{train}}} \subset \mathcal{D}_\nu$. We next compute the associated solutions $\{u(\mu^{(i)}; \nu^{(i)})\}_{i=1}^{N_{\text{train}}}$ to the transformed problem. We then obtain the associated set of principal points for solutions to the non-transformed problem: $\{\{p_j(u(\mu^{(i)}, \nu^{(i)}) \circ g^{-1}(\cdot; \nu^{(i)}))\}_{j=1}^{n_p}\}_{i=1}^{N_{\text{train}}}$, where we recall $u(\mu, \nu) \circ g^{-1}(\cdot; \nu) = \hat{u}(\mu)$. We then solve the optimization problem to best align transformed principal points: for $i = 1, \ldots, N_{\text{train}}$, find

$$\nu^\star(\mu^{(i)}) = \arg\min_{\nu \in \mathcal{D}_\nu} \sum_{j=1}^{n_p} \|g^{-1}(p_j(u(\mu_{\text{ref}}, \nu_{\text{ref}}) \circ g^{-1}(\cdot; \nu_{\text{ref}})); \nu_{\text{ref}}) - g^{-1}(p_j(u(\mu^{(i)}, \nu^{(i)}) \circ g^{-1}(\cdot; \nu^{(i)})); \nu)\|^2_{\ell^2(\mathbb{R}^d)}. \tag{6}$$

We then invoke (5) to construct the map $\xi : \mu \mapsto \nu$ using RBFs and training data $\{\nu^\star(\mu)\}_{\mu \in \mathcal{D}_{\text{train}}}$.

Remark 9. The above two formulations are identical if (i) $\hat{u}(\mu)$ and $u(\mu, \nu)$ are the exact solutions of (1) and (2), respectively, so that $\hat{u}(\mu) = u(\mu, \nu) \circ g^{-1}(\cdot; \nu)$ and (ii) ν_{ref} is chosen to satisfy $g(\cdot; \nu_{\text{ref}}) = I_d$. While the second condition can be satisfied without loss of generality through the choice of transformation parametrization, the first condition cannot be satisfied when the solutions to the PDEs are numerically approximated. As noted above, we use the second formulation in this work, as our NLROM construction procedure naturally works with $u(\mu, \nu)$ instead of $\hat{u}(\mu)$, as we will see in Section 3.3.

Remark 10. Instead of using the difference in the principal point locations $\|p_i(\hat{u}(\mu_{\text{ref}})) - g^{-1}(p_i(\hat{u}(\mu)); \nu)\|$, one could measure, as discussed in the introduction, the difference in the solution fields $\|\hat{u}(\mu_{\text{ref}}) - \hat{u}(\mu) \circ g^{-1}(\cdot; \nu)\|_{\mathcal{Y}}$ or the shock-indicator field $\|\hat{s}^\circ_{\text{dil}}(\hat{u}(\mu_{\text{ref}})) - \hat{s}^\circ_{\text{dil}}(\hat{u}(\mu_{\text{ref}})) \circ g^{-1}(\cdot; \nu)\|_{L^2(\Omega)}$. However, the computation of the difference in the reference field and another arbitrary-transformed field is expensive as it requires the transformation of all quadrature points and the evaluation of the field at those points. If the fields are represented as piecewise polynomials on meshes, then an accurate evaluation of the difference may also require quadrature rules on the intersection of the two meshes. The problem is further exacerbated when the meshes are decomposed for parallel computation, since the transformed quadrature points may lie in a subdomain owned by a different process. For these reasons, we use a simple metric that uses the difference in the principal point locations. That is, by approximating shocks by their principal points, we avoid the cost and complexity of comparing full fields in parallel settings.

Remark 11. Similarly to the above remark, instead of using a shock indicator field $\hat{s}^\circ_{\text{dil}}(u(\mu))$, one could use another field quantity, such as the Mach number field $M(u(\mu))$, to align solution fields. Some of these field sensors are sensitive to features other than shocks. For example, the Mach number field exhibits sharp gradients or discontinuity across not just shocks but also in boundary and shear layers. Using such a sensor provides the potential to not only align shocks but also other features that are difficult to approximate using linear compression schemes, such as shear layers with parameter-dependent locations. This can be

advantageous for viscous flows, and in particular (high-Reynolds-number) RANS flows with thin boundary and shear layers. However, using a field sensor that is sensitive to too many flow features may also cause the optimization algorithm to compromise the alignment of the most important feature: the shocks. In addition, the presence of multiple features can limit the effectiveness of the principal point approximation; e.g., it would be difficult to align both a shock and a boundary layer using the same set of principal points. We hence use the shock indicator field $\hat{s}^{\circ}_{\text{dil}}(u(\mu))$ which focuses solely on the shock regardless of the presence of other features, especially in RANS flows. In other words, for RANS flows, our goal in transformation is to make the decay of the Kolmogorov N-width for transonic problems comparable to that of subsonic flows; we do not aim for further improvement through alignment of other features such as shear layers.

3. Discretization and model reduction of the transformed problem

3.1. Discontinuous Galerkin method on transformed domain

We now introduce a DG approximation of the transformed problem (2). Our treatment of geometry transformation follows the standard "reference domain" or "map-then-discretize" formulation in model reduction; see, e.g., [32, 35]. Our DG formulation for transformed domain specifically follows [13]. We first introduce a triangulation $\mathcal{T}_h$ of the domain Ω, and the associated sets of interior facets $\Sigma \coloneqq \cup_{\kappa\in\mathcal{T}_h}\partial\kappa \setminus \partial\Omega$ and boundary facets $\Gamma \coloneqq \cup_{\kappa\in\mathcal{T}_h}\partial\kappa \cap \partial\Omega$. We then introduce a DG approximation space

$$\mathcal{V}_h \coloneqq \{v \in L^2(\Omega)^m \mid v|_\kappa \in \mathbb{P}^p(\kappa)^m,\ \forall \kappa \in \mathcal{T}_h\}, \tag{7}$$

where $\mathbb{P}^p(\kappa) : \kappa \to \mathbb{R}$ is the space of polynomials of degree at most p on κ. We denote the dimension of the DG space by $N_h = \dim(\mathcal{V}_h)$. We then introduce the symmetric interior penalty (IP) DG discretization of the system of conservation laws [17, 18]. The semilinear form $R : \mathcal{V}_h \times \mathcal{V}_h \times \mathcal{D} \times \mathcal{D}_\nu \to \mathbb{R}$ is given by

$$R(w, v; \mu, \nu) = \int_\Omega r_\Omega(w, v; \mu, \nu)dx + \int_\Sigma r_\Sigma(w, v; \mu, \nu)ds + \int_\Gamma r_\Gamma(w, v; \mu, \nu)ds \quad \forall w, v \in \mathcal{V}, \tag{8}$$

where

$$r_\Omega(w, v; \mu, \nu) \coloneqq \Big[- \nabla_\nu v : F(w; \mu) + \nabla_\nu v : K(w; \mu)\nabla_\nu w - v \cdot S(w, \nabla_\nu w; \mu)\Big]\det(Dg(\cdot; \nu)), \tag{9}$$

$$\begin{aligned} r_\Sigma(w, v; \mu, \nu) \coloneqq \Big[&[v]^+_- \cdot \hat{F}(w^+, w^-; \hat{n}_\nu, \mu) - \{K(w; \mu)\nabla_\nu v\} : [\![w]\!]_\nu \\ &- [\![v]\!]_\nu : \{K(w; \mu)(\nabla_\nu w - \sigma[\![v]\!]_\nu)\}\Big]\det(Dg(\cdot; \nu))\|Dg(\cdot; \nu)^{-T}\hat{n}_0\|_2, \end{aligned} \tag{10}$$

$$\begin{aligned} r_\Gamma(w, v; \mu, \nu) \coloneqq \Big[&v \cdot \hat{F}_\Gamma(w; \hat{n}_\nu, \mu) - K(w; \mu)\nabla_\nu v : ((w - u_\Gamma(w; \mu)) \otimes \hat{n}_\nu) \\ &- (v \otimes \hat{n}_\nu) : G(K(w; \mu)(\nabla_\nu w - \sigma(w - u_\Gamma(w; \mu)) \otimes \hat{n}_\nu); \hat{n}_\nu, \mu)\Big]\det(Dg(\cdot; \nu))\|Dg(\cdot; \nu)^{-T}\hat{n}_0\|_2; \end{aligned} \tag{11}$$

here, $\hat{F} : \mathbb{R}^m \times \mathbb{R}^m \times \mathbb{R}^d \times \mathcal{D} \to \mathbb{R}^m$ is the interior facet numerical flux, $\hat{F}_\Gamma : \mathbb{R}^m \times \mathbb{R}^d \times \mathcal{D} \to \mathbb{R}^m$ is the boundary facet numerical flux, $G : \mathbb{R}^{m\times d} \times \mathbb{R}^d \times \mathcal{D} \to \mathbb{R}^{m\times d}$ is the boundary diffusion flux, $[\![\cdot]\!]$ is the so-called jump operator adopted to transformed domain given by $[\![w]\!]_\nu = w^+ \otimes \hat{n}^+_\nu + w^- \otimes \hat{n}^-_\nu$, $\{\cdot\}$ is the so-called averaging operator given by $\{w\} = (w^+ + w^-)/2$, $\sigma \in \mathbb{R}_{>0}$ is the penalty constant for the IP DG method, and $\otimes$ denotes the tensor product of two first-order tensors to yield a second-order tensor. We note that all of the gradients and normal vectors are associated with the transformed problem (2). Throughout this work, we use Roe's approximate Riemann solver [31] to compute the numerical fluxes. To stabilize the approximation in the presence of shocks, the diffusion term incorporates artificial viscosity based on the local dilation-based shock indicator (3) [26] and the physical Navier-Stokes diffusion model for the Prandtl number of 3/4, which results in the enthalpy of the flow being constant across the shock [30]. Similarly, the output functionals that we consider in this work may be written in terms of its integrands as

$$J(w; \mu, \nu) \coloneqq \int_\Omega j_\Omega(w; \mu, \nu)dx + \int_\Gamma j_\Gamma(w; \mu, \nu)ds \quad \forall w \in \mathcal{V}, \tag{12}$$

where $j_\Omega : \mathcal{V}_h \times \mathcal{D} \times \mathcal{D}_\nu \to L^\infty(\Omega)$ is the volume integrand, and $j_\Gamma : \mathcal{V}_h \times \mathcal{D} \times \mathcal{D}_\nu \to L^\infty(\Gamma)$ is the boundary integrand.

In practice, the integrals over Ω, Σ, and Γ are approximated using an element- and facet-wise Gauss-like quadrature rules. We denote the pairs of quadrature points and weights on Ω, Σ, and Γ by $\{(x_{\Omega,q}, \rho_{\Omega,q})\}_{q=1}^{Q_\Omega}$, $\{(x_{\Sigma,q}, \rho_{\Sigma,q})\}_{q=1}^{Q_\Sigma}$, and $\{(x_{\Gamma,q}, \rho_{\Gamma,q})\}_{q=1}^{Q_\Gamma}$, respectively. The quadrature approximation of the semilinear form (8) is given by

$$\begin{aligned} R_h(w, v; \mu, \nu) &\coloneqq \sum_{q=1}^{Q_\Omega} \rho_{\Omega,q} r_\Omega(w, v; \mu, \nu)(x_{\Omega,q}) + \sum_{q=1}^{Q_\Sigma} \rho_{\Sigma,q} r_\Sigma(w, v; \mu, \nu)(x_{\Sigma,q}) + \sum_{q=1}^{Q_\Gamma} \rho_{\Gamma,q} r_\Gamma(w, v; \mu, \nu)(x_{\Gamma,q}) \\ &\coloneqq \sum_{q=1}^{Q_h^r} \rho_{h,q}^r r_h(w, v; \mu, \nu)(x_{h,q}^r), \end{aligned} \tag{13}$$

where we have introduced a more compact, concatenated notation based on the number of quadrature points $Q_h^r \coloneqq Q_\Omega + Q_\Sigma + Q_\Gamma$, the pairs of quadrature points and weights $\{(x_{h,q}^r, \rho_{h,q}^r)\}_{q=1}^{Q_h^r} = \{(x_{\Omega,q}, \rho_{\Omega,q})\}_{q=1}^{Q_\Omega} \cup \{(x_{\Sigma,q}, \rho_{\Sigma,q})\}_{q=1}^{Q_\Sigma} \cup \{(x_{\Gamma,q}, \rho_{\Gamma,q})\}_{q=1}^{Q_\Gamma}$, and the corresponding mapping of $r_h(\cdot, \cdot; \cdot, \cdot)$ to $r_\Omega(\cdot, \cdot; \cdot, \cdot)$, $r_\Sigma(\cdot, \cdot; \cdot, \cdot)$, and $r_\Gamma(\cdot, \cdot; \cdot, \cdot)$ based on the evaluation point of the integrand. The superscript "r" indicates that the concatenated quadrature-point-wise decomposition is associated with the residual form $R_h(\cdot, \cdot; \cdot, \cdot)$. Similarly, the quadrature approximation of the output functional (12) is given by

$$J_h(w; \mu, \nu) \coloneqq \sum_{q=1}^{Q_\Omega} \rho_{\Omega,q} j_\Omega(w; \mu, \nu)(x_{\Omega,q}) + \sum_{q=1}^{Q_\Gamma} \rho_{\Gamma,q} j_\Gamma(w; \mu, \nu)(x_{\Gamma,q}) \coloneqq \sum_{q=1}^{Q_h^j} \rho_{h,q}^j j_h(w; \mu, \nu)(x_{h,q}^j), \tag{14}$$

where we have again introduced a more compact, concatenated notation based on the number of quadrature points $Q_h^j \coloneqq Q_\Omega + Q_\Gamma$, the pairs of quadrature points and weights $\{(x_{h,q}^j, \rho_{h,q}^j)\}_{q=1}^{Q_h^j} = \{(x_{\Omega,q}, \rho_{\Omega,q})\}_{q=1}^{Q_\Omega} \cup \{(x_{\Gamma,q}, \rho_{\Gamma,q})\}_{q=1}^{Q_\Gamma}$, and the corresponding mapping of $j_h(\cdot, ; \cdot, \cdot)$ to $j_\Omega(\cdot; \cdot, \cdot)$ and $j_\Gamma(\cdot; \cdot, \cdot)$ based on the evaluation point of the integrand, with the superscript "j" indicating the quadrature-point-wise decomposition for the output form $J_h(\cdot; \cdot, \cdot)$.

We now state the (primal) DG problem: given $\mu \in \mathcal{D}$ and $\nu \in \mathcal{D}_\nu$, find $u_h(\mu, \nu) \in \mathcal{V}_h$ such that

$$R_h(u_h(\mu, \nu), v_h; \mu, \nu) = 0 \quad \forall v_h \in \mathcal{V}_h, \tag{15}$$

and then evaluate the output

$$s_h(\mu, \nu) \coloneqq J_h(u_h(\mu, \nu); \mu, \nu).$$

In practice, the (nonlinear) DG problem (15) is solved using a Newton-like method with pseudo-transient continuation (PTC) [22]. In addition, we introduce the dual DG problem, which is required to equip our ROM with an output error estimate: given $\mu \in \mathcal{D}$ and $\nu \in \mathcal{D}_\nu$, find $z_h(\mu, \nu) \in \mathcal{V}_h$ such that

$$R_h'(u_h(\mu, \nu); w_h, z_h(\mu, \nu); \mu, \nu) = J_h'(u_h(\mu, \nu); w_h; \mu, \nu) \quad \forall w_h \in \mathcal{V}_h, \tag{16}$$

where $R_h'(y; w, v; \mu, \nu)$ and $J_h'(y; w; \mu, \nu)$ are the Gateaux derivatives of $R_h(\cdot, v; \mu, \nu)$ and $J_h(\cdot; \mu, \nu)$, respectively, evaluated about y in the direction w. Unlike the primal problem (15), the dual problem (16) is always linear.

Remark 12. Not all DG methods admit a quadrature-point-wise decomposition of the DG residual (13) and output (14). For instance, many of the methods considered in the review paper by Arnold et al. [3], including the popular Bassi and Rebay's second method (BR2) [7], do not admit a point-wise decomposition because they require element-wise lifting, which couples degrees of freedom over each element. Similarly, element-wise shock indicators, such as the spectral indicator by Persson and Peraire [30], do not admit a point-wise decomposition. In this work, we choose the symmetric IP DG method and point-wise dilation-based shock indicator, so that the resulting discretization admits a point-wise decomposition, which is crucial for efficient hyperreduction of high-order methods [13].

3.2. DG error estimation and adaptive mesh refinement

We now equip our DG approximation with an output error estimate and adaptive mesh refinement to control the approximation error. To this end, we use the dual-weighted residual (DWR) method [8]. We first introduce an enriched DG approximation space $\mathcal{V}_h^{\text{en}} \coloneqq \{v \in L^2(\Omega)^m \mid v|_\kappa \in \mathbb{P}^{p+1}(\kappa)^m, \forall \kappa \in \mathcal{T}_h\}$, which comprises polynomials of degree one higher than the space $\mathcal{V}_h$ in (7). We then solve the enriched dual problem: given $\mu \in \mathcal{D}$ and $\nu \in \mathcal{D}_\nu$, find $z_h^{\text{en}}(\mu,\nu) \in \mathcal{V}_h^{\text{en}}$ such that

$$R_h'(u_h(\mu,\nu); w_h, z_h^{\text{en}}(\mu,\nu); \mu, \nu) = J_h'(u_h(\mu,\nu); w_h; \mu, \nu) \quad \forall w_h \in \mathcal{V}_h^{\text{en}}. \tag{17}$$

We then evaluate the DWR error estimate:

$$\eta^{\text{fe}}(\mu,\nu) \coloneqq |R_h(u_h(\mu,\nu), z_h^{\text{en}}(\mu,\nu); \mu, \nu)|, \tag{18}$$

which approximates the output error $|s(\mu) - s_h(\mu,\nu)|$. That is, it approximates the error of the DG output $s_h(\mu,\nu)$ with respect to the exact PDE output $s(\mu)$.

To inform anisotropic adaptive mesh refinement, we also introduce an element-wise error indicator and an anisotropic refinement indicator. Following the standard practice [8, 18], we construct an element-wise error indicator through the element-wise restriction of the error estimate (18):

$$\eta_\kappa^{\text{fe}}(\mu,\nu) \coloneqq |R_h(u_h(\mu,\nu), z_h^{\text{en}}(\mu,\nu)|_\kappa; \mu, \nu)|, \quad \kappa \in \mathcal{T}_h. \tag{19}$$

We then mark the elements with the top α fraction of error for refinement.

Next, to determine *how* to refine those elements, we use an anisotropic refinement indicator based on local solves (cf. [18, 11, 40]). Namely, we first introduce a set of candidate refinement configurations (or triangulations over κ) $\{\mathcal{T}_\kappa^{(1)}, \dots, \mathcal{T}_\kappa^{(n_{\text{config}})}\}$, where n_{config} is the number of candidate refinement configurations. For example, for a rectangular element (in two dimensions), we consider three candidate configurations: anisotropically splitting κ into two elements in one of the element-reference coordinate direction (i.e., "horizontal" split); anistoropically splitting κ into two elements in the other element-reference coordinate direction (i.e., "vertical" split); and isotropically splitting κ into four elements. We then solve the local problem: find $u_\kappa^{(i)}(\mu,\nu) \in \mathcal{V}_\kappa^{(i)}$ such that

$$R_h^{(i)}(u_\kappa^{(i)}(\mu,\nu), v; \mu, \nu) = 0 \quad \forall v \in \mathcal{V}_\kappa^{(i)}, \quad i = 1, \dots, n_{\text{config}}, \tag{20}$$

where $\mathcal{V}_\kappa^{(i)}$ is the piecewise degree-p polynomial space associated with $\mathcal{T}_\kappa^{(i)}$, and $R_h^{(i)}(\cdot,\cdot;\cdot,\cdot)$ is the "localization" of the DG residual form (13) to κ (and hence $\mathcal{T}_\kappa^{(i)}$), which incorporates the Dirichlet boundary condition on $\partial\kappa$ based on $u_h(\mu,\nu)|_{\partial\kappa}$ evaluated on neighboring elements. We then evaluate the elemental error indicator

$$\eta_\kappa^{\text{fe},(i)}(\mu,\nu) \coloneqq |R_h^{(i)}(u_h^{(i)}(\mu,\nu), z_h^{\text{en}}(\mu,\nu)|_\kappa; \mu, \nu)|, \quad i = 1, \dots, n_{\text{config}}. \tag{21}$$

We then find the anisotropic refinement configuration that minimizes the error (indicator) for a given increase in the number of degrees of freedom:

$$i_\kappa^\star = \underset{i \in \{1,\dots,n_{\text{config}}\}}{\arg\min} \frac{|\eta_\kappa^{\text{fe},(i)}(\mu,\nu)|}{\dim(\mathcal{V}_\kappa^{(i)})}. \tag{22}$$

We finally split $\kappa \in \mathcal{T}_h$ according to $\mathcal{T}_\kappa^{(i_\kappa^\star)}$.

Algorithm 1 summarizes the output-based anisotropic adaptive mesh refinement algorithm used in this work. In short, we follow the standard `Solve` $\rightarrow$ `Estimate` $\rightarrow$ `Mark` $\rightarrow$ `Refine` procedure to adaptive mesh refinement, but with an addition of anisotropy detection in the `Estimate` and `Mark` steps, following, e.g., [18, 11].

Algorithm 1: Output-based anisotropic adaptive mesh refinement.

input : Parameters: $\mu \in \mathcal{D}$ and $\nu \in \mathcal{D}_\nu$
Initial mesh: $\mathcal{T}_h$
FE tolerances: δ^{fe}
Refinement fraction: $\alpha \in (0, 1]$

output: Adaptively refined mesh: $\mathcal{T}_h$
Adaptively refined solution: $u_h(\mu, \nu)$

Initialize: set $E = \infty$. **while** $E > \delta^{\text{fe}}$ **do**
  Solve primal problem: solve (15) for $u_h(\mu, \nu)$.
  Solve enriched dual problem: solve (17) for $z_h^{\text{en}}(\mu, \nu)$.
  Compute error estimate: evaluate (18) and set $E \coloneqq \eta^{\text{fe}}(\mu, \nu)$. Terminate if $E < \delta^{\text{fe}}$.
  Compute element-wise error indicators: evaluate (19) for $\{\eta_\kappa^{\text{fe}}(\mu, \nu)\}_{\kappa \in \mathcal{T}_h}$.
  for $\kappa \in \mathcal{T}_h$ *such that* $\eta_\kappa^{\text{fe}}(\mu, \nu)$ *is in top* α *fraction* **do**
    Compute indicator: solve local problems (20) and evaluate error indicators (21).
    Refine element: anisotropically refine element κ according to the optimality indicator (22).
  end
end

3.3. Reduced-order model

We now construct a ROM. To begin, we collect snapshots $U_{\mathcal{D}_{\text{train}},\nu} = \{u_h(\mu^{(i)}, \nu^{(i)})\}_{i=1}^{N_{\text{train}}} \subset \mathcal{U}_{\mathcal{D},\nu}$ associated with a training parameter set $\mathcal{D}_{\text{train}} = \{\mu^{(i)}\}_{i=1}^{N_{\text{train}}}$ and a training transformation parameter set $\mathcal{D}_{\nu,\text{train}} = \{\nu^{(i)}\}_{i=1}^{N_{\text{train}}}$. We then apply proper orthogonal decomposition (POD), if compression is desired, or Gram-Schmidt orthogonalization, if compression is not required, to obtain an RB $\{\phi_i^{\text{pr}}\}_{i=1}^{N}$ for $N \leq N_{\text{train}}$ and in practice $N \ll N_h$; the superscript "pr" indicates that the basis is for the primal (and not dual) problem. We then introduce the RB space $\mathcal{V}_N^{\text{pr}} \coloneqq \text{span}\{\phi_i^{\text{pr}}\}_{i=1}^{N}$. Our RB problem is as follows: given $\mu \in \mathcal{D}$ and $\nu \in \mathcal{D}_\nu$, find $u_N(\mu, \nu) \in \mathcal{V}_N^{\text{pr}}$ such that

$$R_h(u_N(\mu, \nu), v_N; \mu, \nu) = 0 \quad \forall v_N \in \mathcal{V}_N^{\text{pr}}, \tag{23}$$

and then evaluate the output

$$s_N(\mu, \nu) \coloneqq J_h(u_N(\mu, \nu); \mu, \nu). \tag{24}$$

Similarly to the DG problem (15), the (nonlinear) RB problem may be solved using a Newton-like method with PTC.

We also equip the ROM with an output error estimate. To this end, we collect adjoint snapshots $Z_{\mathcal{D}_{\text{train}},\nu} = \{z_h(\mu^{(i)}, \nu^{(i)})\}_{i=1}^{N_{\text{train}}}$. Similarly to the (primal) RB, we then apply POD or Gram-Schmidt to obtain a dual RB $\{\phi_i^{\text{du}}\}_{i=1}^{N^{\text{du}}}$ for $N^{\text{du}} \leq N_{\text{train}}$ and in practice $N^{\text{du}} \ll N_h$, and introduce the associated dual RB space $\mathcal{V}_N^{\text{du}} \coloneqq \text{span}\{\phi_i^{\text{du}}\}_{i=1}^{N^{\text{du}}}$. We then evaluate an RB approximation of the DWR: given $\mu \in \mathcal{D}$ and $\nu \in \mathcal{D}_\nu$, find the dual RB solution $z_N(\mu, \nu) \in \mathcal{V}_N^{\text{du}}$ such that

$$R_h'(u_N(\mu, \nu); w_N, z_N^{\text{du}}(\mu, \nu); \mu, \nu) = J_h'(u_N(\mu, \nu); w_N; \mu, \nu) \quad \forall w_N \in \mathcal{V}_N^{\text{du}}, \tag{25}$$

and evaluate the output error estimate

$$\eta_N^{\text{rb}}(\mu, \nu) \coloneqq |R_h(u_N(\mu, \nu), z_N^{\text{du}}(\mu, \nu); \mu, \nu)|, \tag{26}$$

which approximates the RB output error $|s_h(\mu, \nu) - s_N(\mu, \nu)|$. That is, it approximates the error in the RB output $s_N(\mu, \nu)$ with respect to the DG output $s_h(\mu, \nu)$.

In summary, we can first solve the primal problem (23) and evaluate (24) to obtain the RB output $s_N(\mu, \nu)$. We can then solve the dual problem (25) and evaluate the DWR (26) to equip the RB output $s_N(\mu, \nu)$ with an error estimate $\eta_N^{\text{rb}}(\mu, \nu)$. The evaluation of the solution, output, and error estimate, however, requires $\mathcal{O}(Q_h) = \mathcal{O}(N_h)$ operations since the respective forms involve $\mathcal{O}(Q_h)$ quadrature points. This renders the ROM approximation (without hyperreduction) not online efficient.

3.4. Hyperreduction

We now introduce a hyperreduced approximation of the (primal) RB problem (23), output evaluation (24), dual RB problem (25), and the DWR (26) to enable online-efficient evaluation of the solution, output, and error estimate. To this end, we appeal to the EQP [41] and in particular its variant for point-wise decomposition of DG approximations [13] and for goal-oriented error control [38]. We invoke the EQP to obtain pairs of reduced quadrature points and weights for the (primal) residual form $\{(\tilde{x}_q^r, \tilde{\rho}_q^r)\}_{q=1}^{\tilde{Q}^r}$, the output functional $\{(\tilde{x}_q^j, \tilde{\rho}_q^j)\}_{q=1}^{\tilde{Q}^j}$, and the DWR error estimate $\{(\tilde{x}_q^\eta, \tilde{\rho}_q^\eta)\}_{q=1}^{\tilde{Q}^\eta}$. We endow all variables associated with hyperreduced quantities with $\tilde{\cdot}$. The RQ points are sparse in the sense that $\{\tilde{x}_q^\bullet\}_{q=1}^{\tilde{Q}^\bullet} \subset \{x_q^\bullet\}_{q=1}^{Q_h^\bullet}$ and $\tilde{Q}^\bullet \ll Q_h^\bullet$ for $\bullet \in \{r, j, \eta\}$. The RQ approximation of the residual form used for the primal RB problem (23) and the output functional used for output evaluation (24) are given by

$$\tilde{R}_N(w, v; \mu, \nu) \coloneqq \sum_{q=1}^{\tilde{Q}^r} \tilde{\rho}_q^r r_h(w, v; \mu, \nu)(\tilde{x}_q^r) \quad \text{and} \quad \tilde{J}_N(w, v; \mu, \nu) \coloneqq \sum_{q=1}^{\tilde{Q}^j} \tilde{\rho}_q^j j_h(w; \mu, \nu)(\tilde{x}_q^j).$$

Similarly, the RQ approximation of the residual form and the output functional used for the dual problem (25) and DWR evaluation (26) are given by

$$\tilde{R}_N^{\mathrm{du}}(w, v; \mu, \nu) \coloneqq \sum_{q=1}^{\tilde{Q}^\eta} \tilde{\rho}_q^\eta r_h(w, v; \mu, \nu)(\tilde{x}_q^\eta) \quad \text{and} \quad \tilde{J}_N^{\mathrm{du}}(w, v; \mu, \nu) \coloneqq \sum_{q=1}^{\tilde{Q}^\eta} \tilde{\rho}_q^\eta j_h(w; \mu, \nu)(\tilde{x}_q^\eta).$$

Using these RQ approximations, our RB-RQ solution is $\tilde{u}_N(\mu, \nu) \in \mathcal{V}_N$ such that

$$\tilde{R}_N(\tilde{u}_N(\mu, \nu), v; \mu, \nu) = 0 \quad \forall v \in \mathcal{V}_N, \tag{27}$$

and the associated RB-RQ output is

$$\tilde{s}_N(\mu, \nu) = \tilde{J}_N(\tilde{u}_N(\mu, \nu); \mu, \nu).$$

To equip the RB-RQ output with an online-efficient error estimate, we solve the hyperreduced dual problem for the solution $\tilde{z}_N(\mu, \nu) \in \mathcal{V}_N^{\mathrm{du}}$ such that

$$(\tilde{R}^{\mathrm{du}})'(\tilde{u}_N(\mu, \nu); w_N, \tilde{z}_N^{\mathrm{du}}(\mu, \nu); \mu, \nu) = (\tilde{J}^{\mathrm{du}})'(\tilde{u}_N(\mu, \nu); w_N; \mu, \nu) \quad \forall w_N \in \mathcal{V}_N^{\mathrm{du}}, \tag{28}$$

and evaluate the output error estimate

$$\tilde{\eta}_N^{\mathrm{rb}}(\mu, \nu) \coloneqq |\tilde{R}^{\mathrm{du}}(\tilde{u}_N(\mu, \nu), \tilde{z}_N^{\mathrm{du}}(\mu, \nu); \mu, \nu)|,$$

which approximates the RB-RQ output error $|s_h(\mu, \nu) - \tilde{s}_N(\mu, \nu)|$. That is, it approximates the error in the RB-RQ output $\tilde{s}_N(\mu, \nu)$ with respect to the DG output $s_h(\mu, \nu)$. Similarly to the RB problem (23), we solve the (nonlinear, primal) RB-RQ problem (27) using a Newton-like method with PTC. The dual RB-RQ problem (28), as before, is linear.

3.5. Construction of NLROM

We now outline the construction of NLROM using two algorithms. The first is the simultaneous construction of the adaptively refinement mesh $\mathcal{T}_h$ and the optimized transformation parameter map $\xi : \mathcal{D} \ni \mu \mapsto \nu \in \mathcal{D}_\nu$. The procedure, which follows [34], is summarized in Algorithm 2. The inputs to the algorithm are the training parameter set $\mathcal{D}_{\mathrm{train}} \coloneqq \{\mu^{(i)}\}_{i=1}^{N_{\mathrm{train}}}$ and a sequence of tightening FE tolerances $\delta^{\mathrm{fe},1} \geq \delta^{\mathrm{fe},2} \geq \cdots \geq \delta^{\mathrm{fe},N_a}$, where N_a is the number of adaptation iterations. At the beginning of the adaptation loop, we initialize $\xi^{(j=0)}$ to the zero function: i.e., no geometry transformation. We then compute FE snapshots at training points to construct $\{u_h(\mu^{(i)}, \xi^{(j-1)}(\mu^{(i)}))\}_{i=1}^{N_{\mathrm{train}}}$, adapting the mesh $\mathcal{T}_h$ as necessary to meet the FE tolerance. We then compute the associated optimized transformation parameter set $\{\nu^\star(\mu^{(i)})\}_{i=1}^{N_{\mathrm{train}}}$. We then construct an RBF approximation of ξ. This process is repeated using increasingly tighter FE tolerance until we meet the final tolerance δ^{fe,N_a}.

Algorithm 2: Simultaneous adaptive mesh and optimized map construction.

input : training parameter set $\mathcal{D}_{\text{train}} \coloneqq \{\mu^{(i)}\}_{i=1}^{N_{\text{train}}} \subset \mathcal{D}$
FE tolerances: $\delta^{\text{fe},1}, \dots, \delta^{\text{fe},N_a}$

output: Optimized map: $\xi : \mathcal{D} \ni \mu \mapsto \nu \in \mathcal{D}_\nu$
Adapted mesh: $\mathcal{T}_h$

Initialize: $\xi^{(j=0)} : \mu \mapsto 0$ and $j = 0$
for $j = 1, \dots, N_a$ **do**
  for $i = 1, \dots, N_{\text{train}}$ **do**
    Solve FE problem: solve FE problem (15) for $u_h(\mu^{(i)}, \xi^{(j-1)}(\mu^{(i)}))$; adapt the mesh $\mathcal{T}_h$ as necessary to meet $\delta^{\text{fe},i}$ using Algorithm 1.
    Find optimized transformation parameter: solve optimization problem (6) for $\nu^\star(\mu^{(i)})$.
  end
  Update optimized map: set $\xi^{(j)}(\cdot) = \text{RBF}(\cdot; \{\nu^\star(\mu^{(i)})\}_{i=1}^{N_{\text{train}}})$ following (5).
end
Finalize: set $\xi(\cdot) \coloneqq \xi^{(j)}(\cdot)$.

Remark 13. The simultaneous constructions of an adaptively refined mesh $\mathcal{T}_h$ and optimized map ξ complement each other, as also noted in [34]. First, the error in the shock location estimated using the principal points described in Section 2.4 relative to the true shock location in the (exact) solution $u(\mu)$ of (1) is $\mathcal{O}(h)$, as the error in the solution $u_h(\mu, \nu)$ with respect to $u(\mu)$ is $\mathcal{O}(h)$ and the resolution of the shock indicator $s_{\text{dil}}(u)$ in (3) is also $\mathcal{O}(h)$; hence, the construction of an accurate map $\xi : \mu \mapsto \nu$ requires a commensurately accurate FE solution. Second, an accurate map $\xi : \mu \mapsto \nu$ reduces the region over which the FE mesh has to be refined since the shock remains in one location as the parameter $\mu \in \mathcal{D}$ is varied; this is in contrast to the case without a transformation, which requires the mesh to be refined over the entire region over which the shock appears as the parameter $\mu \in \mathcal{D}$ is varied. Hence, by simultaneously constructing adaptively refined mesh $\mathcal{T}_h$ and optimized map ξ, we obtain an increasingly accurate map while avoiding unnecessary "global" mesh refinement through increasingly accurate shock alignment. In practice, the initial FE tolerance $\delta^{\text{fe},1}$ is set to ∞ so that no mesh adaptation is performed when the shocks are completely unaligned.

The second algorithm, summarized in Algorithm 3 constructs the NLROM. We make an important remark:

Remark 14. Despite considering a nonlinear approximation, the construction follows the standard linear ROM construction procedure: compute the DG snapshots associated with the training parameter set $\mathcal{D}_{\text{train}}$, construct primal and dual RBs using POD, and construct RQ rules using EQP. This is because we have recast the nonlinear approximation of the original problem (1) as a linear approximation of the transformed problem (2). Specifically, given an optimized parameter map $\xi : \mathcal{D} \ni \mu \to \nu \in \mathcal{D}_\nu$, we are "simply" solving the problem associated with a μ-parametrized semilinear form $\mathcal{D} \ni \mu \mapsto r(\cdot, \cdot; \mu, \xi(\mu))$. Hence, the construction of the NLROM follows the standard linear ROM construction procedure.

4. Examples

4.1. Transonic Euler flow over NACA–Gaussian bump

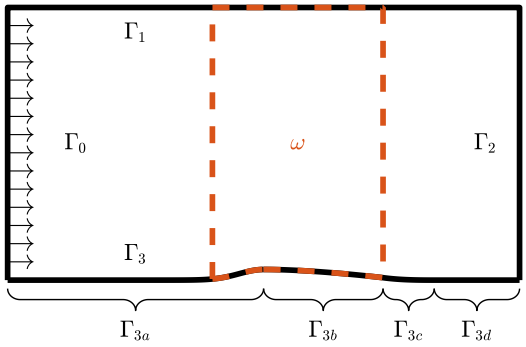
We first consider transonic inviscid flow over a bump composed from a Gaussian and the NACA0012 profile. The geometry, shown in Figure 3a, is as follows: the boundary $\partial\Gamma_{3a}$ for $x_1 \in [-1.5, 0]$ is a Gaussian bump of a standard deviation 0.2 and a height 0.0625; the boundary $\partial\Gamma_{3b}$ for $x_1 \in [0, 0.7]$ is a NACA0012 such that the bottom boundary is C^1 continuous with the Gaussian bump at $x_1 = 0$; the boundary $\partial\Gamma_{3c}$ for $x_1 \in [0.7, 1.0]$ is a quartic spline such that the bottom boundary is C^2 continuous at $x = 0.7$ and $x = 1.0$;

Algorithm 3: NLROM construction.

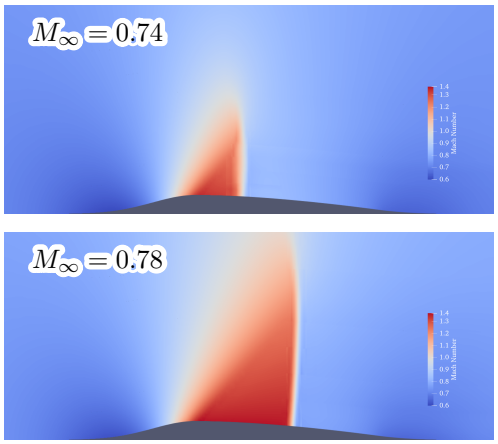
input : Optimized map: $\xi : \mathcal{D} \ni \mu \mapsto \nu \in \mathcal{D}_\nu$
training set $\mathcal{D}_{\text{train}} \subset \mathcal{D}$
RB tolerance: δ^{rb}
EQP tolerances: δ^r, δ^J, δ^η

output: RBs: $\{\phi_i^{\text{pr}}\}_{i=1}^{N}$ and $\{\phi_i^{\text{du}}\}_{i=1}^{N^{\text{du}}}$
RQs: $\{\tilde{x}_q^r, \tilde{\rho}_q^r\}_{q=1}^{\tilde{Q}^r}$, $\{\tilde{x}_q^j, \tilde{\rho}_q^j\}_{q=1}^{\tilde{Q}^j}$, and $\{\tilde{x}_q^\eta, \tilde{\rho}_q^\eta\}_{q=1}^{\tilde{Q}^\eta}$

for $i = 1, \ldots, N_{\text{train}}$ **do**
  Solve primal and dual FE problem: solve primal FE problem (15) for $u_h(\mu^{(i)}, \xi(\mu^{(i)}))$ and dual FE problem (16) for $z_h(\mu^{(i)}, \xi(\mu^{(i)}))$ on adapted mesh $\mathcal{T}_h$
end
Construct primal RB: apply POD to obtain primal RB: $\{\phi_i^{\text{pr}}\}_{i=1}^{N} = \text{POD}_{\delta^{\text{rb}}}(\{u_h(\mu^{(i)}, \xi(\mu^{(i)}))\}_{i=1}^{N_{\text{train}}})$
Construct dual RB: apply POD to obtain dual RB: $\{\phi_i^{\text{du}}\}_{i=1}^{N^{\text{du}}} = \text{POD}_{\delta^{\text{rb}}}(\{z_h(\mu^{(i)}, \xi(\mu^{(i)}))\}_{i=1}^{N_{\text{train}}})$
Construct RQs: apply EQP to obtain RQs $\{\tilde{x}_q^\bullet, \tilde{\rho}_q^\bullet\}_{q=1}^{\tilde{Q}^\bullet}$ for $\bullet \in \{r, j, \eta\}$.



(a) geometry.

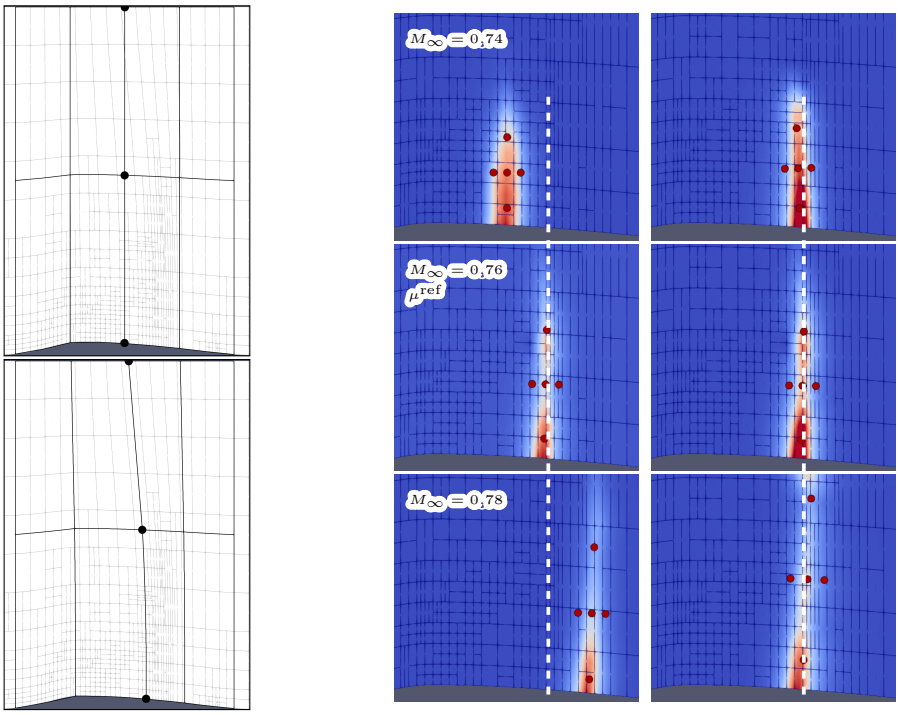


(b) Mach number fields

Figure 3: The geometry and Mach number fields, for the two extreme parameter values, of the NACA–Gauss bump Euler problem.

and $\partial\Gamma_{3d}$ for $x_1 \in [1.0, 1.5]$ is flat. The height of the domain is 1.6. The governing equation is the two-dimensional compressible Euler equations in entropy variables [6]. The parameter is the freestream Mach number $M_\infty \in [0.74, 0.78] := \mathcal{D}$. We specify the stagnation temperature, stagnation pressure, and angle of attack (i.e., 0°) on the inflow boundary Γ_0. We specify the static pressure on the outflow boundary Γ_2. These boundary conditions are chosen to yield the desired freestream Mach number. All other boundaries are slip walls. The quantity of interest is the drag, which takes on a value in $[7.25 \times 10^{-3}, 1.951 \times 10^{-2}]$ over $\mathcal{D}$. The solution for the two extreme values of the parameter is shown in Figure 3b; we observe that the shock location and size are sensitive to the small change in the freestream Mach number, which is also reflected in the large change in the drag value.

We define the transformation region $\omega \subset \Omega$ as shown in Figure 3a. This is the region in the domain where shocks are expected based on solutions for extreme parameter values. We introduce a 5×3 grid shown in Figure 4a to induce geometry transformation. As discussed in Section 2.3, the five nodes in the flow direction are necessary to meet the regularity conditions at the interface of ω and $\Omega \setminus \omega$. (We need not have five nodes in vertical direction as ω spans the entire height of Ω and there is no top or bottom interface.) We parametrize the transformation by the flow-wise displacement of the three middle nodes, allowing sliding



(a) grid and control points

(b) shock indicator and principal points

Figure 4: Illustration of transformation for the NACA–Gauss bump Euler problem. Subfigure (a) shows grid and control points for untransformed (top) and transformed (bottom) configurations; active control points are marked with black dots. Subfigure (b) shows shock indicator and principal points for untransformed (left) and transformed (right) configurations; dotted lines show alignment.

on the bottom boundary Γ_{3b}. An illustration of the transformed grid is shown in Figure 4a. We use the principal point approximation of the dilation shock sensor, which for $d = 2$ dimensions has $n_p = 2d + 1 = 5$ points. An example of the shock sensor, principal points, and alignment of shocks via transformation is illustrated in Figure 4b.

To construct the ROM and NLROM, we compute DG snapshot sets $\widehat{U}_{\mathcal{D}} = \{\hat{u}_h(\mu^{(i)})\}_{i=1}^{N_{\text{train}}}$ and $U_{\mathcal{D},\xi} = \{u_h(\mu^{(i)}, \xi(\mu^{(i)}))\}_{i=1}^{N_{\text{train}}}$, respectively, for $N_{\text{train}} = 15$ parameters equispaced in $\mathcal{D}$. For ROM, we adapt the mesh so that each DG snapshot meets the drag error tolerance of 5×10^{-5}, which is approximately 0.5%. For NLROM, we use Algorithm 2 with four outer iterations at target DG drag error tolerances of $\{1 \times 10^{-3}, 1 \times 10^{-4}, 5 \times 10^{-5}, 5 \times 10^{-5}\}$ to simultaneously adapt the mesh and optimize the transformation. We equip both the ROM and NLROM with EQP-based hyperreduction and DWR-based error estimation.

We compare the output error convergence for ROM and NLROM in Figure 5. The reported error is the maximum over the randomly selected test set $\Xi_{\text{test}} = \{\mu_{\text{test}}^{(i)}\}_{i=1}^{N_{\text{test}}}$ of size $N_{\text{test}} = 5$ so that $\mu_{\text{test}}^{(i)} \notin \mathcal{D}_{\text{train}}$. The NLROM error decreases rapidly with N, converging to below 0.5% with only $N = 7$ modes, while the ROM error is greater than 10% using all $N = 15$ modes. The linear approximation space of the ROM suffers from the slow decay of the Kolmogorov N-width; the proposed nonlinear approximation strategy based on transformation mitigates the issue to provide rapid convergence. Specifically, while the shock alignment achieved by the proposed formulation using a shock indicator, principle points, and spline-based transformation is never perfect, it is sufficient to enable rapid output error convergence. The NLROM convergence "stagnates" after meeting the target drag error level (i.e., for $N > 7$), as the NLROM error becomes comparable to the error in DG snapshots; we recall that the fidelity of the transformation and hence the shock alignment is limited by the DG solution resolution, and hence we cannot expect rapid error convergence below the DG snapshot error level.

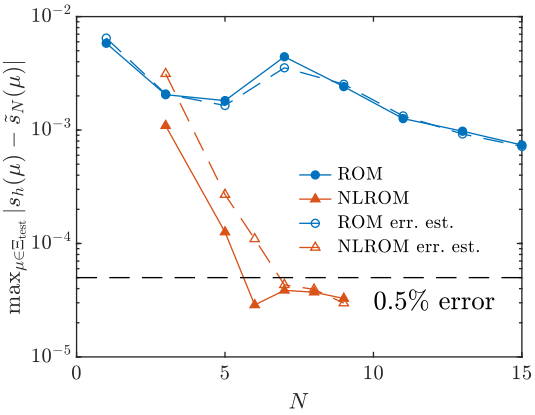


Figure 5: Output error convergence for linear ROM and NLROM for the NACA–Gauss bump Euler problem.

Table 1: Summary of linear ROM and NLROM for the NACA–Gauss bump Euler problem.

	full-order model : reduced-order model			c_d error	speedup
	$N_h : N$	$Q_h : Q$	$t_{\text{FOM}} : t_{\text{ROM}}$ [s]	$\lvert s_h(\mu) - \tilde{s}_N(\mu)\rvert$	$t_{\text{FOM}}/t_{\text{ROM}}$
ROM	21216 : 15	43404 : 427	73.8 : 0.0143	$> 10\%$	5160
NLROM	16344 : 7	33504 : 483	21.9 : 0.0130	$< 0.5\%$	5680

Figure 5 also shows that the DWR error estimate, while it is not a bound, accurately predicts the error with respect to the DG "truth" approximation for both ROM and NLROM in practice. Specifically, the DWR error estimate correctly informs the user that the ROM is in fact not working. This ability to know when, and more importantly when not to, trust ROM predictions is crucial to confidently use ROMs in predictive settings.

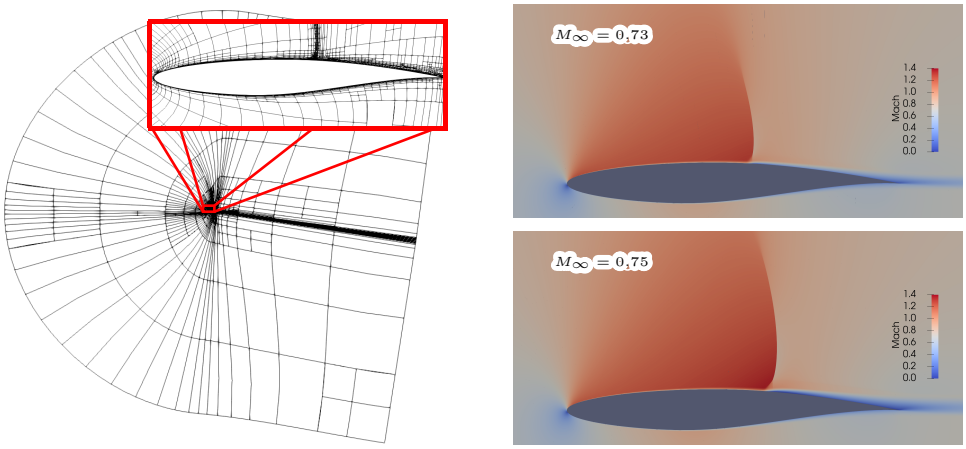
Table 1 provides key attributes of the ROM and NLROM. First, we note that the number of degrees of freedom of DG snapshots is $N_h = 21216$ for the ROM and $N_h = 16344$ for the NLROM. In the ROM, as the DG approximation uses the same (non-transformed) mesh to compute all snapshots, the mesh has to be refined over the entire range of shock locations to meet the drag error tolerance of 0.5% for all parameter values. On the other hand, NLROM uses the simultaneous mesh adaptation and transformation procedure described in Algorithm 2 to obtain (transformed) snapshots whose shocks are aligned and hence requires fewer degrees of freedom to meet the same drag error tolerance.

Second, we note that hyperreduction yields RQ rules with $\tilde{Q} = 427$ points for ROM and $\tilde{Q} = 483$ points for NLROM; both are a significant reduction from $Q_h = 43404$ and $Q_h = 33504$. Figure 6 shows the difference in the RQ points between ROM and NLROM. We note that, in ROM, the RQ points are spread in the wide shock region. On the other hand, in NLROM, the RQ points are concentrated in the reference solution shock region and the front region of the supersonic pocket. This indicates, as expected, that fewer RQ points are required to accurately capture the shock, since only those in the reference shock location are required to approximate the residual for all parameter values. However, the transformation also induces (undesired) transformation in the front region of the supersonic pocket, where the adjoint exhibits rapid change, which results in the introduction of more RQ points. The additional RQ points in the front region of the supersonic pocket are not required when using a larger transformation region ω relative to the shock, as in the RAE2822 example which follows.

Third, we note in Table 1 that ROM and NLROM yield a speedup (i.e., $t_{\text{FOM}}/t_{\text{ROM}}$) of approximately 5160 and 5680 times, respectively, with respect to a single DG solve on the (non-transformed) mesh associated with the ROM. The ROM and NLROM computational time is dominated by the residual and Jacobian evaluation, which scales with the number of RQ points $\tilde{Q}$, and hence the two methods having a similar

(a) ROM (b) NLROM

Figure 6: RQ points superimposed on the solution for the reference parameter value for the NACA–Gauss bump Euler problem.



(a) computational domain (b) Mach number field

Figure 7: The computational domain and the Mach number fields, for the two extreme parameter values, for the RAE2822 RANS-SA problem.

online time (t_{ROM}) is expected. However, we recall that ROM drag error exceeds 10% while the NLROM drag error is below 0.5%, and hence the NLROM is much more efficient (in terms of accuracy per cost) than ROM.

4.2. Transonic RANS-SA flow over RAE2822

We next consider fully turbulent transonic flow over the RAE2822 airfoil. The governing equation is the two-dimensional compressible RANS equations with the Spalart–Allmaras (SA) turbulence model with the SA-neg fix [33, 1]. The parameter is the freestream Mach number $M_\infty \in [0.73, 0.75] := \mathcal{D}$. The angle of attack and the Reynolds number are fixed at $\alpha = 2.75°$ and $Re_c = 6.5 \times 10^6$, respectively. The quantity of interest is the drag, which takes on a value in $[1.70 \times 10^{-2}, 2.63 \times 10^{-2}]$ over $\mathcal{D}$. The solution for the two extreme values of the parameter is shown in in Figure 7b; we again observe that the shock location is sensitive to the small change in the freestream Mach number, which is also reflected in the large change in the drag value.

We define $\omega \subset \Omega$ on the top surface of the airfoil, where we expect the shocks, as shown in Figure 8. In this case, ω does not extend to the top of the domain, and we accordingly require sufficient number of grid nodes in both vertical and horizontal directions to satisfy the regularity conditions of our transformation. The typical height of the shocks in the range $M_\infty \in [0.73, 0.75]$ is $\mathcal{O}(0.5c)$ where c is the chord length of the

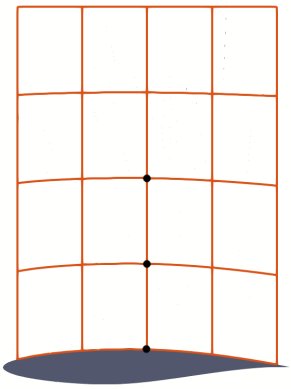
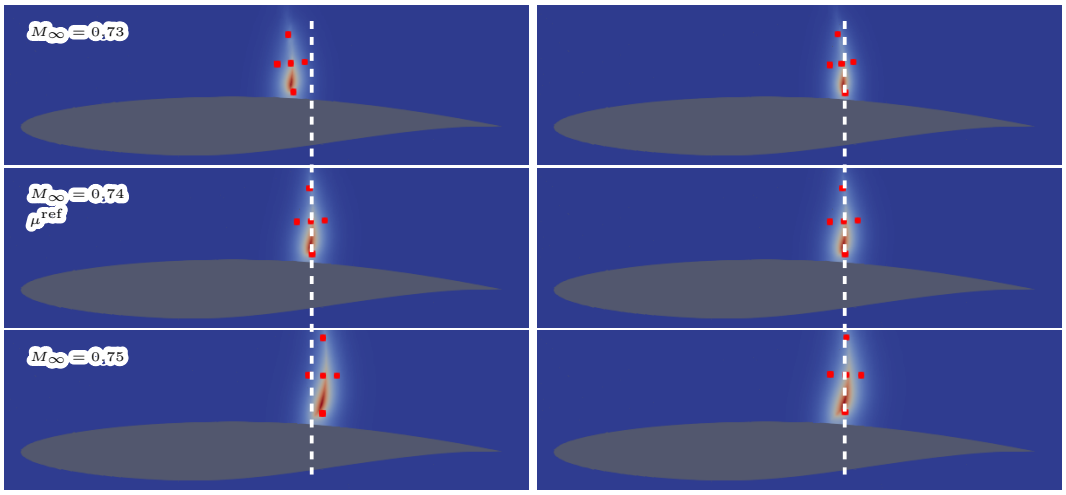


Figure 8: Transformation subdomain ω and control points for the RAE2822 RANS-SA problem. Black dots mark active control points.



(a) non-transformed solutions (b) transformed solutions

Figure 9: Non-transformed (a) and transformed (b) shock indicator $\hat{s}^\circ_{\text{dil}}(u)$ and principal points for $M_\infty \in \{0.73, 0.74, 0.75\}$ for the RAE2822 RANS-SA problem. Dotted lines show alignment to the reference solution at $M_\infty = 0.74$.

airfoil. To that end, we select ω to have a height of $1.5c$ and width of $0.9c$. We then define a 5×5 grid of nodes shown in Figure 8 and parametrize the transformation using the horizontal displacement of the three central nodes. We again use the normalized dilation shock sensor and principal point approximation, which yields $n_p = 5$ principal points. An example of the transformation aligning shocks using the principal points is illustrated in Figure 9.

To construct ROM and NLROM, we compute DG snapshot sets $\widehat{U}_{\mathcal{D}} = \{\hat{u}_h(\mu^{(i)})\}_{i=1}^{N_{\text{train}}}$ and $U_{\mathcal{D},\xi} = \{u_h(\mu^{(i)}, \xi(\mu^{(i)}))\}_{i=1}^{N_{\text{train}}}$, respectively, for $N_{\text{train}} = 20$ parameters equispaced in $\mathcal{D}$. As before, we adaptively refine the mesh to meet the target DG drag error tolerance of 5×10^{-5}, which is $\approx 0.5\%$. For NLROM, we invoke Algorithm 3, with four iterations of mesh adaptation and transformation with error tolerances $\{1{\times}10^{-3}, 5{\times}10^{-4}, 5{\times}10^{-5}, 5{\times}10^{-5}\}$. We equip both the ROM and NLROM with EQP-based hyperreduction and DWR-based error estimation.

Figure 10 shows the error convergence of ROM and NLROM. We report the maximum error over the randomly selected test set $\Xi_{\text{test}} = \{\mu^{(i)}_{\text{test}}\}_{i=1}^{N_{\text{test}}}$ for $N_{\text{test}} = 8$ so that $\mu^{(i)}_{\text{test}} \notin \mathcal{D}_{\text{train}}$. Errors are evaluated against DG solutions. We observe slow error convergence for ROM, with the error fluctuating at around 3% for $N = 10$–16 and converging below 0.5% for $N = 20$, and rapid convergence for NLROM, with the error

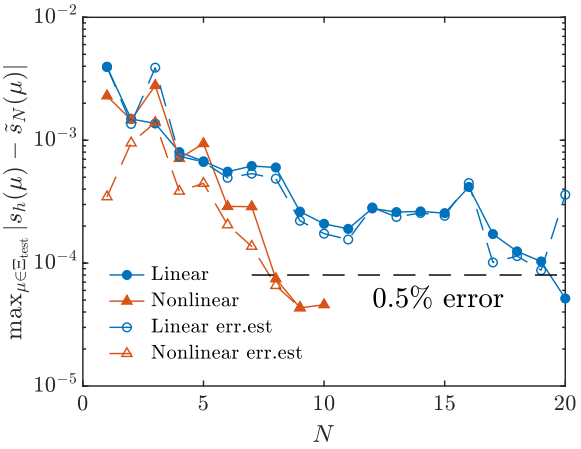


Figure 10: Error convergence of linear ROM and NLROM for the RAE2822 RANS-SA problem.

converging to below 0.5% for $N = 9$. The ROM, which uses a linear approximation space, again suffers from the slow decay of the Kolmogorov N-width for this transonic flow. The proposed nonlinear approximation strategy mitigates the issue to provide rapid convergence. We emphasize that the transformation strategy is effective even for this transonic RANS flow, which exhibits multiple sharp features that are not shocks, including the boundary and shear layers, in both mean flow states and the Spalart–Allmaras turbulence model.

Figure 10 also shows that the error estimate accurately approximates the true error with respect to the DG solutions. The error estimate is again effective when ROM/NLROM approximations are accurate and, arguably more importantly, when it is not accurate. We emphasize that the DWR error estimate remains effective despite the significant nonlinearity introduced by both the shocks and the Spalart–Allmaras turbulence model.

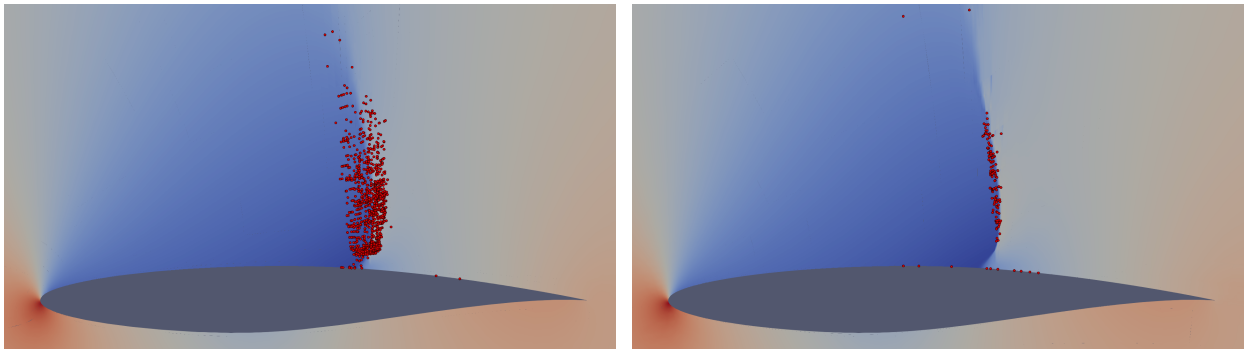
Table 2 summarizes key attributes of the ROM and NLROM. First, we note that the DG snapshots have $N_h = 84030$ degrees of freedom for ROM and $N_h = 81150$ for NLROM. The impact of simultaneous mesh adaptation and transformation is smaller than observed for the Euler bump case, as a large portion of the degrees of freedom are used to resolve the boundary and shear layers. However, the transformation results in a significant difference in the efficiency of hyperreduction: $\tilde{Q} = 698$ RQ points for ROM compared to $\tilde{Q} = 210$ RQ points for NLROM. This is due to two factors: first, the size of the RB is much smaller for NLROM than for ROM ($N = 9$ vs $N = 20$); second, the ROM must introduce RQ points over the entire range of the shock motion, whereas the NLROM need RQ points only in the aligned shock region, as shown in Figure 11. Unlike in the NACA–Gauss bump problem, the range of shock locations is narrower, the deformations required to align shocks are smaller, and no additional RQ points are introduced in the front region of the supersonic pocket. The threefold reduction in the number of RQ points yields a commensurate speedup of NLROM compared to ROM; the speedup for the NLROM is $\approx 8,000$ at the 0.5% error level. While both ROM and NLROM are able to approximate solutions to 0.5% accuracy, the NLROM needs a smaller RB to achieve this. In addition, the accuracy of the ROM is in part an artifact of the artificial viscosity introduced in the diffusion term of (8), which smears the shock and makes it easier to approximate; if we refine the snapshots further to achieve smaller than 0.5% drag error, the ROM will require a larger RB to provide accurate solutions.

5. Summary and perspectives

This work develops and assesses a registration-based NLROM method for efficient model reduction of parametrized PDEs that exhibit slowly decaying Kolmogorov N-widths. The proposed NLROM is composed

Table 2: Summary of linear ROM and NLROM for the RAE2822 RANS-SA problem.

	full-order model : reduced-order model			c_d error	speedup
	$N_h : N$	$Q_h : Q$	$t_{\text{FOM}} : t_{\text{ROM}}$ [s]	$\lvert s_h(\mu) - \tilde{s}_N(\mu)\rvert$	$t_{\text{FOM}}/t_{\text{ROM}}$
ROM	84030 : 20	137874 : 698	$0.64 \times 10^3 : 2.27 \times 10^{-1}$	$< 0.5\%$	2819
NLROM	81150 : 9	132876 : 210	$0.71 \times 10^3 : 8.08 \times 10^{-2}$	$< 0.5\%$	7920



(a) ROM (b) NLROM

Figure 11: RQ points superimposed on the solution for the reference parameter value for the RAE2822 RANS-SA problem.

of the following ingredients: (i) an arbitrarily transformable PDE and discretization; (ii) a parametrized composite transformation, which uses transfinite interpolation to map a subdomain ω to the unit domain and spline transformations to warp points inside a unit domain; (iii) a dilation-based shock indicator and principal point approximation of this shock indicator to compute optimal transformation parameters; (iv) online efficient interpolation of transformation parameters; (v) EQP-based hyperreduction and DWR-based error estimates; and (vi) simultaneous mesh adaptation and transformation to generate snapshots. These ingredients are combined to yield an NLROM, which approximates the solution in a transformed manifold $\mathcal{U}_{\mathcal{D},\nu} = \{u_h(\mu, \nu \equiv \xi(\mu))\}_{\mu\in\mathcal{D}}$ with aligned shocks and hence a rapidly decaying Kolmogorov N-width.

The proposed NLROM provides the following features: (i) it provides efficient transformations suitable for many transonic flows, particularly those over an airfoil, where the shock is localized to a small region; (ii) it is scalable to parallel settings due to efficient shock sensor approximation using principal points (unlike approaches that require full-field matching and hence the solution to a distributed and arbitrary mesh projection problem); (iii) it does not resort to extensive offline training to achieve online accuracy (unlike typical neural-network based approaches); (iv) it is equipped with hyperreduction and online-efficient error estimates; and (v) it has been demonstrated on two-dimensional inviscid Euler *and* transonic RANS problems, the latter of which exhibit significant nonlinearities not just in shocks but also in the Spalart–Allmaras turbulence model.

The future work includes the extension of the NLROM to three-dimensional problems as well as to problems that exhibit multiple shocks and hence requires multiple and/or more flexible transformation.

Acknowledgments

The financial support for this work was provided by the Natural Sciences and Engineering Research Council of Canada. Some of the computations were performed on the Niagara supercomputer at the SciNet HPC Consortium. SciNet is funded by the Canada Foundation for Innovation; the Government of Ontario; Ontario Research Fund - Research Excellence; and the University of Toronto.

References

[1] S. R. Allmaras, F. T. Johnson, and P. R. Spalart. Modifications and clarifications for the implementation of the Spalart-Allmaras turbulence model. 7th International Conference on Computational Fluid Dynamics ICCFD7-1902, ICCFD, 2012.

[2] D. Amsallem, M. J. Zahr, and C. Farhat. Nonlinear model order reduction based on local reduced-order bases. *International Journal for Numerical Methods in Engineering*, 92(10):891–916, jun 2012.

[3] D. N. Arnold, F. Brezzi, B. Cockburn, and L. D. Marini. Unified analysis of discontinuous Galerkin methods for elliptical problems. *SIAM Journal on Numerical Analysis*, 39(5):1749–1779, 2002.

[4] J. Barnett and C. Farhat. Quadratic approximation manifold for mitigating the Kolmogorov barrier in nonlinear projection-based model order reduction. *Journal of Computational Physics*, 464:111348, sep 2022.

[5] J. Barnett, C. Farhat, and Y. Maday. Neural-network-augmented projection-based model order reduction for mitigating the Kolmogorov barrier to reducibility. *Journal of Computational Physics*, 492:112420, nov 2023.

[6] T. J. Barth. Numerical methods for gasdynamic systems on unstructured meshes. In D. Kröner, M. Ohlberger, and C. Rohde, editors, *An Introduction to Recent Developments in Theory and Numerics for Conservation Laws*, pages 195–282. Springer-Verlag, 1999.

[7] F. Bassi, S. Rebay, G. Mariotti, S. Pedinotti, and M. Savini. A high-order accurate discontinuous finite element method for inviscid and viscous turbomachinery flows. In R. Decuypere and G. Dibelius, editors, *Turbomachinery - Fluid Dynamics and Thermodynamics, European Conference, 2*, pages 99–108, 1997.

[8] R. Becker and R. Rannacher. An optimal control approach to a posteriori error estimation in finite element methods. *Acta Numerica*, 10:1–102, may 2001.

[9] P. Benner, S. Gugercin, and K. Willcox. A survey of projection-based model reduction methods for parametric dynamical systems. *SIAM Review*, 57(4):483–531, 2015.

[10] N. Cagniart, Y. Maday, and B. Stamm. Model order reduction for problems with large convection effects. In *Contributions to Partial Differential Equations and Applications*, pages 131–150. Springer International Publishing, 2019.

[11] M. Ceze and K. J. Fidkowski. Anisotropic hp-adaptation framework for functional prediction. *AIAA Journal*, 51(2):492–509, 2012.

[12] D. S. Ching, P. J. Blonigan, F. Rizzi, and J. A. Fike. Model reduction of hypersonic aerodynamics with residual minimization techniques. In *AIAA SCITECH 2022 Forum*. American Institute of Aeronautics and Astronautics, jan 2022.

[13] E. Du and M. Yano. Efficient hyperreduction of high-order discontinuous Galerkin methods: Element-wise and point-wise reduced quadrature formulations. *Journal of Computational Physics*, 466:111399, oct 2022.

[14] J. L. Eftang, A. T. Patera, and E. M. Rønquist. An "*hp*" certified reduced basis method for parametrized elliptic partial differential equations. *SIAM Journal on Scientific Computing*, 32(6):3170–3200, jan 2010.

[15] A. Ferrero, T. Taddei, and L. Zhang. Registration-based model reduction of parameterized two-dimensional conservation laws. *Journal of Computational Physics*, 457:111068, may 2022.

[16] S. Fresca and A. Manzoni. POD-DL-ROM: Enhancing deep learning-based reduced order models for nonlinear parametrized PDEs by proper orthogonal decomposition. *Computer Methods in Applied Mechanics and Engineering*, 388:114181, jan 2022.

[17] R. Hartmann and P. Houston. Symmetric interior penalty DG methods for the compressible Navier-Stokes equations I: method formulation. *International Journal of Numerical Analysis & Modeling*, 3(1):1–20, 2006.

[18] R. Hartmann and P. Houston. Error estimation and adaptive mesh refinement for aerodynamic flows. In H. Deconinck, editor, *VKI LS 2010-01: 36th CFD/ADIGMA course on hp-adaptive and hp-multigrid methods, Oct. 26-30, 2009*. Von Karman Institute for Fluid Dynamics, Rhode Saint Genèse, Belgium, 2009.

[19] J. S. Hesthaven, G. Rozza, and B. Stamm. *Certified reduced basis methods for parametrized partial differential equations*. Springer, 2016.

[20] A. Iollo and D. Lombardi. Advection modes by optimal mass transfer. *Physical Review E*, 89(2), feb 2014.

[21] A. Iollo and T. Taddei. Mapping of coherent structures in parameterized flows by learning optimal transportation with Gaussian models. *Journal of Computational Physics*, 471:111671, dec 2022.

[22] C. T. Kelley and D. E. Keyes. Convergence analysis of pseudo-transient continuation. *SIAM Journal on Numerical Analysis*, 35(2):508–523, apr 1998.

[23] Y. Kim, Y. Choi, D. Widemann, and T. Zohdi. A fast and accurate physics-informed neural network reduced order model with shallow masked autoencoder. *Journal of Computational Physics*, 451:110841, feb 2022.

[24] K. Lee and K. T. Carlberg. Model reduction of dynamical systems on nonlinear manifolds using deep convolutional autoencoders. *Journal of Computational Physics*, 404:108973, 2020.

[25] M. A. Mirhoseini and M. J. Zahr. Model reduction of convection-dominated partial differential equations via optimization-based implicit feature tracking. *Journal of Computational Physics*, 473:111739, Jan. 2023.

[26] D. Moro, N. C. Nguyen, and J. Peraire. Dilation-based shock capturing for high-order methods. *International Journal for Numerical Methods in Fluids*, 82(7):398–416, 2016.

[27] N. J. Nair and M. Balajewicz. Transported snapshot model order reduction approach for parametric, steady-state fluid flows containing parameter-dependent shocks. *International Journal for Numerical Methods in Engineering*, 117(12):1234–1262, 2019.

[28] M. Ohlberger and S. Rave. Nonlinear reduced basis approximation of parameterized evolution equations via the method of freezing. *Comptes Rendus Mathematique*, 351(23-24):901–906, dec 2013.

[29] M. Ohlberger and S. Rave. Reduced basis methods: success, limitations and future challenges. In *Proceedings of the Conference Algoritmy*, pages 1–12, 2016.

[30] P.-O. Persson and J. Peraire. Sub-cell shock capturing for discontinuous Galerkin methods. AIAA 2006-0112, AIAA, 2006.

[31] P. L. Roe. Approximate Riemann solvers, parameter vectors, and difference schemes. *Journal of Computational Physics*, 43(2):357–372, 1981.

[32] G. Rozza, D. B. P. Huynh, and A. T. Patera. Reduced basis approximation and a posteriori error estimation for affinely parametrized elliptic coercive partial differential equations — Application to transport and continuum mechanics. *Archives of Computational Methods in Engineering*, 15(3):229–275, 2008.

[33] P. R. Spalart and S. R. Allmaras. A one-equation turbulence model for aerodynamics flows. *La Recherche Aérospatiale*, 1:5–21, 1994.

[34] T. Taddei. A registration method for model order reduction: data compression and geometry reduction. *SIAM Journal on Scientific Computing*, 42(2):A997–A1027, jan 2020.

[35] T. Taddei and L. Zhang. A discretize-then-map approach for the treatment of parameterized geometries in model order reduction. *Computer Methods in Applied Mechanics and Engineering*, 384:113956, oct 2021.

[36] T. Taddei and L. Zhang. Registration-based model reduction in complex two-dimensional geometries. *Journal of Scientific Computing*, 88(3), Aug. 2021.

[37] G. Welper. Interpolation of functions with parameter dependent jumps by transformed snapshots. *SIAM Journal on Scientific Computing*, 39(4):A1225–A1250, jan 2017.

[38] M. Yano. Goal-oriented model reduction of parametrized nonlinear partial differential equations: Application to aerodynamics. *International Journal for Numerical Methods in Engineering*, 121(23):5200–5226, jun 2020.

[39] M. Yano. Model reduction in computational aerodynamics. In *Model Order Reduction*, pages 201–236. De Gruyter, dec 2020.

[40] M. Yano and D. L. Darmofal. An optimization-based framework for anisotropic simplex mesh adaptation. *Journal of Computational Physics*, 231(22):7626–7649, 2012.

[41] M. Yano and A. T. Patera. An LP empirical quadrature procedure for reduced basis treatment of parametrized nonlinear PDEs. *Computer Methods in Applied Mechanics and Engineering*, 344:1104–1123, feb 2019.