Registration-based model reduction of parameterized PDEs with spatio-parameter adaptivity

Nicolas Barral, Tommaso Taddei, Ishak Tifouti

Registration-based model reduction of parameterized PDEs with spatio-parameter adaptivity

Nicolas Barral[1], **Tommaso Taddei**[2], **Ishak Tifouti**[1]

[1] *Univ. Bordeaux, CNRS, Bordeaux INP, IMB, UMR 5251, F-33400 Talence, France*
Inria Bordeaux Sud-Ouest, Team CARDAMOM, 33400 Talence, France,
nicolas.barral@inria.fr,ishak.tifouti@inria.fr

[2] *Univ. Bordeaux, CNRS, Bordeaux INP, IMB, UMR 5251, F-33400 Talence, France*
Inria Bordeaux Sud-Ouest, Team MEMPHIS, 33400 Talence, France, tommaso.taddei@inria.fr

Abstract

We propose an automated nonlinear model reduction and mesh adaptation framework for rapid and reliable solution of parameterized advection-dominated problems, with emphasis on compressible flows. The key features of our approach are threefold: (i) a metric-based mesh adaptation technique to generate an accurate mesh for a range of parameters, (ii) a general (i.e., independent of the underlying equations) registration procedure for the computation of a mapping Φ that tracks moving features of the solution field, and (iii) an hyper-reduced least-square Petrov-Galerkin reduced-order model for the rapid and reliable estimation of the mapped solution. We discuss a general paradigm — which mimics the refinement loop considered in mesh adaptation — to simultaneously construct the high-fidelity and the reduced-order approximations, and we discuss actionable strategies to accelerate the offline phase. We present extensive numerical investigations for a quasi-1D nozzle problem and for a two-dimensional inviscid flow past a Gaussian bump to display the many features of the methodology and to assess the performance for problems with discontinuous solutions.

Keywords: parameterized conservation laws; model order reduction; mesh adaptation; registration methods; nonlinear approximations.

1 Introduction

1.1 Lagrangian model reduction of steady conservation laws

In the past few decades, there has been an increasing demand for rapid and reliable reduced-order models (ROMs) for many-query and real-time applications such as design optimization, uncertainty quantification, real-time control and monitoring. Despite the many contributions to the field, model order reduction of advection-dominated partial differential equations (PDEs) remains a formidable task that requires major improvements of state-of-the-art procedures. The goal of this paper is to devise an integrated model order reduction (MOR) mesh adaptation (MA) procedure for nonlinear advection-dominated PDEs: our approach combines projection-based MOR, mesh adaptation and registration techniques to simultaneously build a parsimonious yet accurate high-fidelity (HF) discretization, a low-rank representation of the solution field that depends on a modest number of generalized coordinates, and a ROM that can be rapidly solved for new values of the parameters.

We consider PDE problems that depend on a vector of P parameters. We denote by μ the vector of model parameters in the region $\mathcal{P} \subset \mathbb{R}^P$; we denote by $\Omega \subset \mathbb{R}^d$ the open computational domain[1]; given the parametric field $w : \Omega \times \mathcal{P} \to \mathbb{R}$, we also introduce notation $w_\mu := w(\cdot;\mu) : \Omega \to \mathbb{R}$. Given $\mu \in \mathcal{P}$, we denote by $q_\mu^{\text{true}} : \Omega \to \mathbb{R}^D$ the vector of D state variables that satisfies the hyperbolic conservation law:

$$\nabla \cdot F_\mu(q_\mu^{\text{true}}) = S_\mu(q_\mu^{\text{true}}) \quad \text{in } \Omega, \tag{1}$$

where $F : \mathbb{R}^D \times \mathcal{P} \to \mathbb{R}^{D\times d}$ is the physical flux and $S : \mathbb{R}^D \times \mathcal{P} \to \mathbb{R}^D$ is the source term. We further introduce the Hilbert space $\mathcal{X} := [L^2(\Omega)]^D$ endowed with the inner product $(\cdot,\cdot)$ and the induced norm $\|\cdot\| := \sqrt{(\cdot,\cdot)}$, such that $(w,v) = \int_\Omega w \cdot v \, dx$ for all $w, v \in \mathcal{X}$; we define the solution manifold $\mathcal{M} = \{q_\mu^{\text{true}} : \mu \in \mathcal{P}\} \subset \mathcal{X}$ that collects the solutions to (1) for all parameter values in the prescribed parameter range. We denote by $\mathcal{T}_{\text{hf}} = \left(\{x_j^{\text{hf}}\}_{j=1}^{N_{\text{nd}}}, \mathtt{T}\right)$ a mesh of the domain Ω with nodes $\{x_j^{\text{hf}}\}_j$ and connectivity matrix $\mathtt{T}$ (see section 2); given the bijection $\Phi : \Omega \to \mathbb{R}^d$, we use notation $\Phi(\mathcal{T}_{\text{hf}})$ to refer to the mesh with deformed nodes $\{\Phi(x_j^{\text{hf}})\}_j$ and the same connectivity $\mathtt{T}$ as $\mathcal{T}_{\text{hf}}$.

[1] To simplify the presentation, in the introduction we assume that the domain does not depend on the parameters; however, in the numerical examples, we shall consider the case of parameterized geometries.

As extensively discussed in the MOR literature, effective model reduction of advection-dominated PDEs is extremely challenging for state-of-the-art procedures. First, the vast majority of MOR methods rely on linear or affine approximations, that is

$$q_\mu^{\text{true}} \approx \widehat{q}_\mu^{\text{lin}} = \mathtt{Z}\widehat{\boldsymbol{\alpha}}_\mu, \tag{2}$$

where $\mathtt{Z} : \mathbb{R}^n \to \mathcal{X}$ is a linear or affine operator, and $\widehat{\boldsymbol{\alpha}} : \mathcal{P} \to \mathbb{R}^n$ is a function of the parameter — in the MOR literature, $\mathtt{Z}$ is typically dubbed *reduced-order basis* (ROB), while $\widehat{\boldsymbol{\alpha}}_\mu$ are referred to as *generalized coordinates*. As shown in several studies (e.g., [36]), linear methods are fundamentally ill-suited to deal with parameter-dependent sharp gradients that naturally arise in the solutions to conservation laws of the form (1). Second, MOR methods typically rely on a single HF mesh to describe all elements of the solution manifold $\mathcal{M}$. For advection-dominated problems, MA is of paramount importance for computational tractability. However, if parametric variations strongly affect the location of sharp-gradient regions, we are forced to refine the mesh over a vast portion of the domain Ω, which leads to HF discretizations of intractable size. Effective MOR procedures for conservation laws should thus embed an effective parametric MA strategy to track moving structures.

The provable inadequacy of linear ansatzs (2) for conservation laws has motivated the development of several nonlinear approximation methods [1, 3, 27, 38]; a promising class of nonlinear approximations is given by *Lagrangian methods* [13, 23, 33, 34, 35, 45, 43] based on the ansatz

$$\widehat{q}_\mu = \widetilde{q}_\mu \circ \Phi_\mu^{-1}, \quad \text{where } \widetilde{q}_\mu = \mathtt{Z}\widehat{\boldsymbol{\alpha}}_\mu, \;\; \Phi_\mu = \mathtt{N}\left(\widehat{\mathbf{a}}_\mu\right). \tag{3}$$

As in (2), $\mathtt{Z} : \mathbb{R}^n \to \mathcal{X}$ is a linear (or affine) operator, and $\widehat{\boldsymbol{\alpha}} : \mathcal{P} \to \mathbb{R}^n$ is a vector-valued function of generalized coordinates; on the other hand, $\mathtt{N} : \mathbb{R}^m \to \mathrm{Lip}(\Omega; \mathbb{R}^d)$ is a suitable, possibly nonlinear, operator that is informed by the domain Ω and $\widehat{\mathbf{a}} : \mathcal{P} \to \mathbb{R}^m$ is a vector-valued function of generalized coordinates for the mapping.

Lagrangian approaches are motivated by the observation (see, e.g., [24] and [43]) that for many problems in continuum mechanics coherent structures that are troublesome for linear approximations — such as shear layers, wakes, shocks and cracks — vary smoothly with the parameter. The mapping $\Phi : \Omega \times \mathcal{P} \to \Omega$ in (3) should hence be designed to track moving features of the solution field and ultimately improve the compressibility of the mapped solution manifold $\widetilde{\mathcal{M}} = \{\widetilde{q}_\mu^{\text{true}} := q_\mu^{\text{true}} \circ \Phi_\mu : \mu \in \mathcal{P}\}$. The task of finding the mapping Φ based on approximate snapshots of the solution manifold $\mathcal{M}$ is referred to as *registration problem* [43]. Note that, by tracking sharp features of the solution field, registration facilitates also the task of building a common mesh for all elements of the (mapped) solution manifold: the mapping Φ hence provides a systematic way to perform parameter-dependent r-adaptivity [9, 32].

1.2 Adaptive construction of Lagrangian reduced-order models

In this paper, we propose a general paradigm for the simultaneous construction of the HF and reduced-order approximations, which mimics the refinement loop considered in MA. The general procedure is sketched in Algorithm 1. Given an initial mesh $\mathcal{T}_{\text{hf}}^{(0)}$ of Ω and the training set $\mathcal{P}_{\text{train}} = \{\mu^k\}_{k=1}^{n_{\text{train}}} \subset \mathcal{P}$, our method returns an HF mesh $\mathcal{T}_{\text{hf}}$, a low-rank mapping Φ, a ROB $\mathtt{Z}$ and a ROM for the generalized coordinates $\widehat{\boldsymbol{\alpha}}$ (cf. (3)) based on an iterative procedure that comprises four distinct steps.

1. Snapshot generation: $(\mathcal{T}_{\text{hf}}, \Phi, \mathcal{P}_{\text{train}}) \to \left\{q_\mu^{\text{hf}} : \mu \in \mathcal{P}_{\text{train}}\right\}$. We generate snapshots of the solution field for all values of the parameter μ in $\mathcal{P}_{\text{train}}$ based on the parametric mesh $\mu \mapsto \Phi_\mu(\mathcal{T}_{\text{hf}})$.

2. Mesh adaptation: $\left\{\widetilde{q}_\mu^{\text{hf}} := q_\mu^{\text{hf}} \circ \Phi_\mu : \mu \in \mathcal{P}_{\text{train}}\right\} \to \mathcal{T}_{\text{hf}}$. We exploit the available set of snapshots to generate an accurate yet parsimonious mesh for the elements of the mapped manifold $\widetilde{\mathcal{M}}$.

3. Registration: $\left(\left\{q_\mu^{\text{hf}} : \mu \in \mathcal{P}_{\text{train}}\right\}, \;\; \mathcal{T}_{\text{hf}}\right) \to \Phi$. We exploit the available set of snapshots to find a parametric mapping Φ that tracks coherent, parameter-dependent structures of the solution field. The method should ensure that the deformed mesh $\Phi_\mu(\mathcal{T}_{\text{hf}})$ is a proper mesh of Ω for all $\mu \in \mathcal{P}$.

4. Linear-subspace model reduction: $(\mathcal{T}_{\text{hf}}, \Phi, \mathcal{P}_{\text{train}}) \to (\mathtt{Z}, \mathrm{ROM})$. We apply linear-subspace MOR to determine the low-rank expansion $\mu \mapsto \widetilde{q}_\mu$, that is we build the ROB $\mathtt{Z}$ and the ROM for $\mu \mapsto \widehat{\boldsymbol{\alpha}}_\mu$.

Algorithm 1 : adaptive training procedure.

1: Initialization: define the mesh $\mathcal{T}_{\text{hf}}^{(0)} = \mathcal{T}_{\text{hf}}^{(1)}$, the mapping $\Phi^{(0)} = \texttt{id}$ (identity map), and the training set $\mathcal{P}_{\text{train}} = \{\mu^k\}_{k=1}^{n_{\text{train}}} \subset \mathcal{P}$.
2: **for** $k = 1, \dots, N_{\text{it}}$ **do**
3: Snapshot generation $\left(\mathcal{T}_{\text{hf}}^{(k-1)}, \Phi^{(k-1)}, \mathcal{P}_{\text{train}}\right) \to \left\{q_\mu^{\text{hf},(k)} \,:\, \mu \in \mathcal{P}_{\text{train}}\right\}.$
4: **if** $k > 1$ **then**
5: Mesh adaptation (cf. section 3) $\left\{q_\mu^{\text{hf},(k)} \circ \Phi_\mu^{(k-1)} : \mu \in \mathcal{P}_{\text{train}}\right\} \to \mathcal{T}_{\text{hf}}^{(k)}.$
6: **end if**
7: Registration (cf. section 4) $\left(\left\{q_\mu^{\text{hf},(k)} \,:\, \mu \in \mathcal{P}_{\text{train}}\right\},\ \mathcal{T}_{\text{hf}}^{(k)}\right) \to \Phi^{(k)}.$
8: Linear-subspace model reduction (cf. section 5) $\left(\mathcal{T}_{\text{hf}}^{(k)},\ \Phi^{(k)},\ \mathcal{P}_{\text{train}}\right) \to \left(\mathtt{Z}^{(k)}, \text{ROM}^{(k)}\right).$
9: **end for**

Similarly to the standard MA loop, our method relies on multiple iterations to address the inaccuracy of the HF estimates at early iterations. We show that the iterative procedure in Algorithm 1 can be significantly accelerated using information from previous iterations (cf. section 6).

The outline of the paper is as follows. In section 2, we introduce relevant notation and the two model problems considered for numerical assessment. In sections 3, 4, and 5, we discuss the problems of mesh adaptation, registration, and model reduction; in section 6 we discuss how to accelerate the training procedure by exploiting information from previous iterations; in section 7, we present extensive numerical investigations to illustrate the effectiveness of our approach. Section 8 concludes the paper.

1.3 Contributions and relation to previous works

This paper extends the work in [20] in several ways: first, we propose an adaptive, iterative procedure for the simultaneous construction of the HF mesh, the mapping Φ, and the reduced-order approximation for the mapped field; second, we incorporate an automated parametric mesh adaptation strategy that is directly informed by the estimated solution fields; third, we discuss viable new strategies to accelerate the training procedure. As in [20], the building blocks of Algorithm 1 exploit methodologies from previous works. The registration procedure was first proposed in [43] and then extended in [47, 48], while we rely on a projection-based least-squares Petrov-Galerkin (LSPG, [10, 11]) ROM with empirical test space chosen as in [48], hyper-reduction based on a variant of the mesh-sampling/empirical quadrature procedures first proposed in [18, 54], and *discretize-then-map* treatment of geometry variations (cf. [15, 46, 52]). We also observe that the idea of using ROMs and/or HF models of variable fidelity at training stage to reduce training costs has been explored in [19, 25].

We rely on a metric-based approach ([29, 30]) to mesh adaptation. Given the HF field q_μ^{hf}, we compute the Hessian of the Mach number in the reference configuration to determine the metric $\mathfrak{M}_\mu$ for the parameter $\mu \in \mathcal{P}_{\text{train}}$; then, we resort to metric intersection [4, 5] to devise a common metric for the entire snapshot set. In this work, we rely on the open-source mesh adaptation toolkit `mmg2d` ([2, 16]) to generate adapted meshes from a (possibly anisotropic) metric $\mathfrak{M}$.

As discussed in section 4, the mapping coefficients $\widehat{\mathbf{a}}_\mu$ in (3) are computed using a non-intrusive (regression) approach, while the solution generalized coordinates $\widehat{\boldsymbol{\alpha}}_\mu$ are computed using projection; on the other hand, Mirhoseini and Zahr in [33] have recently proposed a coupled Lagrangian MOR approach to simultaneously learn solution and mapping generalized coordinates. As opposed to [33], our choice enables the use of standard projection-based linear-subspace MOR methods for parameterized geometries: it hence has the potential to achieve faster online predictions and much easier integration with existing HF and MOR routines, possibly at the price of larger offline training costs.

Several authors have proposed to include (parametric) mesh adaptation procedures in the MOR framework. Simultaneous adaptivity in space and in parameter — in effect, spatio-parameter adaptivity — was proposed by Yano in [53] and further developed in [42]. The approaches in [42, 53] exploit h-MA and weak-greedy sampling of the parameter space, and rely on the explicit instantiation of a *super-mesh* over the entire parameter domain; the size of the super-mesh might hence be prohibitively large for advection-dominated problems. To address this issue, Little and Farhat have proposed in [28] to combine h-MA with clustering in parameter domain — more precisely in state space — to avoid the explicit definition of a super-mesh that is valid for all parameters. The present work represents the first attempt to systematically combine parametric r-MA (induced by the mapping) with parameter-independent h-MA, in the MOR framework.

We finally remark that several authors have considered ansatzs of the form

$$\widehat{q}_\mu = \widetilde{q}_\mu \circ \Phi_\mu, \quad \text{where} \quad \widetilde{q}_\mu : \mathbb{R}^d \to \mathbb{R}^D, \quad \Phi_\mu : \Omega \to \mathbb{R}^d, \tag{4}$$

with $\widetilde{q}_\mu, \Phi_\mu$ possibly nonlinear low-rank operators (see, e.g., [7, 26]). Note that, unlike in (3), Φ_μ is not necessarily a bijection from Ω in itself; note also that the field $\widetilde{q}_\mu$ needs to be defined over $\mathbb{R}^d$. As shown in [26], approximations of the form (4) can potentially handle shock topology changes; on the other hand, we remark that (4) is inherently nonlinear and hence requires the development of specialized projection techniques.

2 Problem statement and finite element discretization

We consider the problem of approximating the solution to the parameterized Euler equations; we consider the equations in non-dimensional form. We refer to [49] for a thorough introduction to the mathematical model and to its physical interpretation. We denote by ρ the fluid density, by u the velocity field, by E the total energy and by p the (static) pressure; we consider the following relationship between pressure and conserved variables:

$$p(q) = (\gamma - 1)\left(E - \frac{1}{2}\rho\|u\|_2^2\right), \tag{5a}$$

where γ is the ratio of specific heats, which is set equal to 1.4. We further introduce the speed of sound a, the Mach number Ma, the total temperature T_{tot}, the total pressure p_{tot} and the total enthalpy H_{tot} such that

$$\begin{aligned} &a = \sqrt{\gamma\frac{p}{\rho}}, \;\; \text{Ma} = \frac{\|u\|_2}{a}, \;\; T = \frac{p}{R\rho}, \;\; T_{\text{tot}} = T\left(1 + \frac{\gamma-1}{2}\text{Ma}^2\right), \\ &p_{\text{tot}} = p\left(1 + \frac{\gamma-1}{2}\text{Ma}^2\right)^{\frac{\gamma-1}{\gamma}}, H_{\text{tot}} = \frac{E+p}{\rho}, \;\; R = \gamma - 1. \end{aligned} \tag{5b}$$

We introduce the finite element (FE) mesh $\mathcal{T}_{\text{hf}} = \left(\{x_j^{\text{hf}}\}_{j=1}^{N_{\text{hf}}}, \mathtt{T}\right)$ of the domain $\Omega \subset \mathbb{R}^d$: the points $\{x_j^{\text{hf}}\}_{j=1}^{N_{\text{nd}}} \subset \overline{\Omega}$ are the nodes of the mesh, the matrix $\mathtt{T} \in \mathbb{N}^{n_{\text{lp}}, N_{\text{e}}}$ is the connectivity matrix where n_{lp} is the number of degrees of freedom in each element and N_{e} is the number of elements. We define the reference element $\widehat{\mathtt{D}} = \{x \in (0,1)^d : \sum_{i=1}^d x_i < 1\}$, the space $\mathbb{P}_{\mathtt{p}}(\widehat{\mathtt{D}})$ of polynomials of degree lower or equal to $\mathtt{p}$, the Lagrangian basis $\{\ell_i\}_{i=1}^{n_{\text{lp}}}$ of the polynomial space $\mathbb{P}_{\mathtt{p}}(\widehat{\mathtt{D}})$ associated with the nodes $\{\tilde{x}_i\}_{i=1}^{n_{\text{lp}}} \subset \overline{\widehat{\mathtt{D}}}$; then, we define the elements $\{\mathtt{D}_k\}_{k=1}^{N_{\text{e}}}$ as the images of the FE maps $\Psi_k^{\text{hf}} : \widehat{\mathtt{D}} \to \mathtt{D}_k$ such that

$$\Psi_k^{\text{hf}}(\tilde{x}) = \sum_{i=1}^{n_{\text{lp}}} x_{\mathtt{T}_{i,k}}^{\text{hf}} \, \ell_i(\tilde{x}), \quad k = 1, \dots, N_{\text{e}}. \tag{6}$$

We also define the FE space associated with the mesh $\mathcal{T}_{\text{hf}}$,

$$\mathcal{X}_{\text{hf}} = \left\{v \in [L^2(\Omega)]^D \; : \; v \circ \Psi_k^{\text{hf}} \in [\mathbb{P}_{\mathtt{p}}(\widehat{\mathtt{D}})]^D, \;\; k = 1, \dots, N_{\text{e}}\right\}, \tag{7}$$

where $D = d + 2$ corresponds to the number of state variables. If $u \in \mathcal{X}_{\text{hf}}$, we denote by $\mathbf{u} \in \mathbb{R}^{N_{\text{hf}}}$ the corresponding FE vector such that

$$u\Big|_{\mathtt{D}_k} = \sum_{i=1}^{n_{\text{lp}}} \sum_{\ell=1}^{D} (\mathbf{u})_{\mathtt{I}_{i,k,\ell}} \;\; \ell_{i,k} \;\; e_\ell, \qquad k = 1, \dots, N_{\text{e}}, \;\; \mathtt{I}_{i,k,\ell} = i + (k-1)n_{\text{lp}} + (\ell - 1)n_{\text{lp}}N_{\text{e}}, \tag{8}$$

where $e_1, \dots, e_d$ are the vectors of the canonical basis of $\mathbb{R}^D$, and $N_{\text{hf}} = D \cdot n_{\text{lp}} \cdot N_{\text{e}}$.

In view of the FE formulation, we introduce the facets $\mathcal{F}_{\text{hf}} = \{\mathtt{F}_j\}_{j=1}^{N_{\text{f}}}$ of the mesh: for each facet, we denote by $\mathbf{n}^+$ the positive normal[2] to the facet and we define the element $\mathtt{D}_j^+$ that contains $\mathtt{F}_j$ and whose normal on $\mathtt{F}_j$ is equal to $\mathbf{n}^+$; for internal facets, we also define the element $\mathtt{D}_j^-$ such that $\mathtt{D}_j^+ \cap \mathtt{D}_j^- = \mathtt{F}_j$. We also define the restriction operators $E_k : \mathcal{X}_{\text{hf}} \to [L^2(\mathtt{D}_k)]^D$ and $E_j^\pm : \mathcal{X}_{\text{hf}} \to [L^2(\mathtt{D}_j^\pm)]^D$ such that $E_k u = u|_{\mathtt{D}_k}$ and $E_j^\pm u = u|_{\mathtt{D}_j^\pm}$ for $k = 1, \dots, N_{\text{e}}$ and $j = 1, \dots, N_{\text{f}}$.

Remark 2.1. *Exploiting* (8)*, we find that any FE field $u \in \mathcal{X}_{\text{hf}}$ is uniquely characterized by the pair $(\mathcal{T}_{\text{hf}}, \mathbf{u})$. Given the bijection $\Phi : \Omega \to \mathbb{R}^d$, we introduce the deformed mesh $\Phi(\mathcal{T}_{\text{hf}}) = \left(\{\Phi(x_j^{\text{hf}})\}_{j=1}^{N_{\text{hf}}}, \mathtt{T}\right)$, and the corresponding FE maps $\{\Psi_{k,\Phi}^{\text{hf}}\}_k$ and FE space $\mathcal{X}_{\text{hf},\Phi}$: it is easy to verify that if $(\Phi(\mathcal{T}_{\text{hf}}), \mathbf{u})$ interpolates the field u in the nodes of $\Phi(\mathcal{T}_{\text{hf}})$, then $(\mathcal{T}_{\text{hf}}, \mathbf{u})$ interpolates $u \circ \Phi$ in the nodes of $\mathcal{T}_{\text{hf}}$. This implies that the ansatz* (3) *can be stored as*

$$\mu \in \mathcal{P} \mapsto (\Phi_\mu(\mathcal{T}_{\text{hf}}), \mathbf{Z}\widehat{\boldsymbol{\alpha}}_\mu) \tag{9}$$

where $\mathbf{Z} \in \mathbb{R}^{N_{\text{hf}} \times n}$ is a parameter-independent matrix.

[2]The choice of the positive normal is arbitrary for internal facets and coincides with the outward normal to Ω for boundary facets.

2.1 Finite element formulation

We consider a discontinuous Galerkin (DG) FE formulation of the compressible Euler equations. We rely on a Laplacian artifical viscosity model (see, e.g., [39]) based on the piecewise-constant dilation-based viscosity:

$$\nu|_{\mathtt{D}_k} \;=\; c_\nu \left(\frac{h_k}{\mathtt{p}}\right)^2 \int_{\mathtt{D}_k} (-\nabla\cdot u)_+ \, dx, \quad \text{with } \; h_k = |\mathtt{D}_k|^{1/d}, \;\; c_\nu > 0. \tag{10}$$

We refer to [57] for a thorough review of artificial viscosity models for DG formulations. We consider the local Lax-Friedrichs (Rusanov) convective flux, and symmetric interior penalty diffusive flux. If we denote by q the vector of state variables, the DG formulation of the conservation law (1) can be stated as follows: find $q^{\mathrm{hf}} \in \mathcal{X}_{\mathrm{hf}}$ such that

$$\mathfrak{R}^{\mathrm{hf}}(q^{\mathrm{hf}}, v) = \sum_{k=1}^{N_{\mathrm{e}}} r_k^{\mathrm{e}}(E_k q^{\mathrm{hf}}, E_k v) \;+\; \sum_{j=1}^{N_{\mathrm{f}}} r_j^{\mathrm{f}}(E_j^+ q^{\mathrm{hf}}, E_j^+ q^{\mathrm{hf}}, E_j^+ v, E_j^- v), \quad \forall\, v \in \mathcal{X}_{\mathrm{hf}}; \tag{11a}$$

where the elemental residuals r_k^{e} are given by

$$r_k^{\mathrm{e}}(q, v) = \int_{\mathtt{D}_k} (-F(q) + \nu(q)\nabla q) : \nabla v \;-\; S(q)\cdot v \, dx, \quad k = 1, \ldots, N_{\mathrm{e}}, \tag{11b}$$

while the facet residuals r_j^{f} are given by

$$r_j^{\mathrm{f}}(q, v) = \int_{\mathtt{F}_j} \mathfrak{H}(q, \mathbf{n}^+)\cdot J(v)\, dx - \int_{\mathtt{F}_j \setminus \partial\Omega} \big\{\nu(q)\nabla q \mathbf{n}^+\big\}\cdot J(v) + \big\{\nu(q)\nabla v \mathbf{n}^+\big\}\cdot J(q) - \frac{\gamma_{\mathrm{ip}}}{|\mathtt{F}_j|} J(q)\cdot J(v)\, dx, \tag{11c}$$

for $j = 1, \ldots, N_{\mathrm{f}}$. Here, $q^{\pm}(x) = \lim_{\epsilon\to 0^+} q(x \mp \epsilon \mathbf{n}^+)$, $J(v) = v^+ - v^-$ and $\{v\} = \frac{1}{2}(v^+ + v^-)$ if $x \notin \partial\Omega$, while $J(v) = \{v\} = v$ if $x \in \partial\Omega$; finally, $\mathfrak{H}(q, \mathbf{n}^+)$ is the convective numerical flux which embeds the definition of the boundary conditions and for which we omit the explicit expression (see, e.g., [21, Appendix B] for the details).

In the numerical experiments, we consider polynomials of degree $\mathtt{p} = 2$; we set $c_\nu = 0.1$ for the model problem of section 2.2.1 and $c_\nu = 10$ for the model problem of section 2.2.2; on the other hand, we consider $\gamma_{\mathrm{ip}} = 10\mathtt{p}^2$. Finally, we solve the discrete problem (11) using the pseudo-transient continuation (PTC) strategy discussed in [56]; in the absence of prior information about the solution field, we initialize the iterative procedure with the free-stream solution.

2.2 Model problems

2.2.1 Inviscid flow through a nozzle

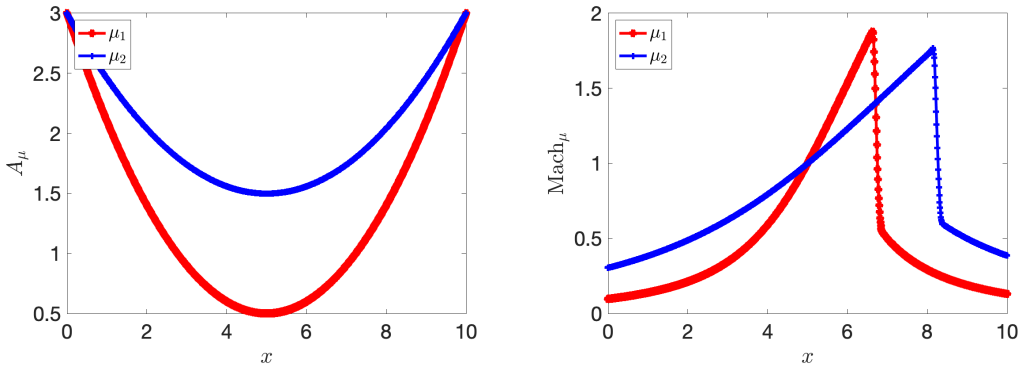
We study the inviscid transonic flow of an ideal gas through a converging-diverging duct. We define the domain $\Omega = (0, L)$, the area $A : \Omega \to \mathbb{R}_+$, the state q, the flux F and the source term S such that

$$q = \begin{bmatrix} A\rho \\ A\rho u \\ AE \end{bmatrix}, \quad F(q) = \begin{bmatrix} A\rho u \\ A(\rho u^2 + p) \\ Au(E+p) \end{bmatrix}, \quad S(q) = \begin{bmatrix} 0 \\ p\partial_x A \\ 0 \end{bmatrix}, \quad A(x) = 3 + 4(A_0 - 3)\frac{x}{L}\left(1 - \frac{x}{L}\right). \tag{12a}$$

Then, we consider the conservation law:

$$\partial_x F(q^{\mathrm{true}}) = S(q^{\mathrm{true}}) \quad x \in \Omega, \tag{12b}$$

completed with a subsonic inlet condition where we prescribe total pressure $p_{\mathrm{tot}} = 0.95$ and total temperature $T_{\mathrm{tot}} = 0.95$, and a subsonic outlet condition where we prescribe the static pressure p_0. Note that the free-stream field is uniquely determined by the data $p_{\mathrm{tot}}, T_{\mathrm{tot}}, p_0$ through (5). We set $L = 10$; furthermore, we consider the parameter vector $\mu = [A_0, p_0]$ in the region $\mathcal{P} = [0.5, 1.5] \times [0.7, 0.85]$. Figures 1(a) and (b) show the behavior of the area throat and the Mach number for two parameter values.



(a) (b)

Figure 1: inviscid flow through a nozzle. (a)-(b) throat area and Mach field for $\mu_1 = [0.5, 0.7]$ and $\mu_2 = [1.5, 0.7]$.

2.2.2 Inviscid flow over a Gaussian bump

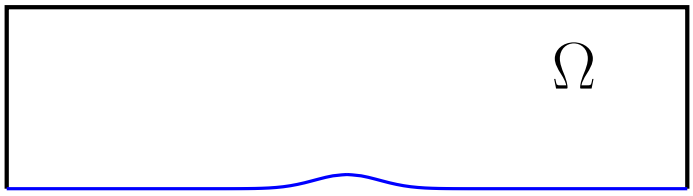
We also consider the two-dimensional inviscid flow past a Gaussian bump. We introduce the domain $\Omega = \{x \in (-1.5, 1.5) \times (0, 0.8) : x_2 > he^{-25x_1^2}\}$ where $h > 0$ is a given parameter (cf. Figure 2(a)). We consider the conservation law:

$$\nabla \cdot F(q^{\text{true}}) = 0, \quad \text{where } q^{\text{true}} = \begin{bmatrix} \rho^{\text{true}} \\ \rho^{\text{true}} u^{\text{true}} \\ E^{\text{true}} \end{bmatrix}, \quad F(q) = \begin{bmatrix} \rho u^\top \\ \rho u u^\top \\ (E+p)u^\top \end{bmatrix}, \tag{13}$$

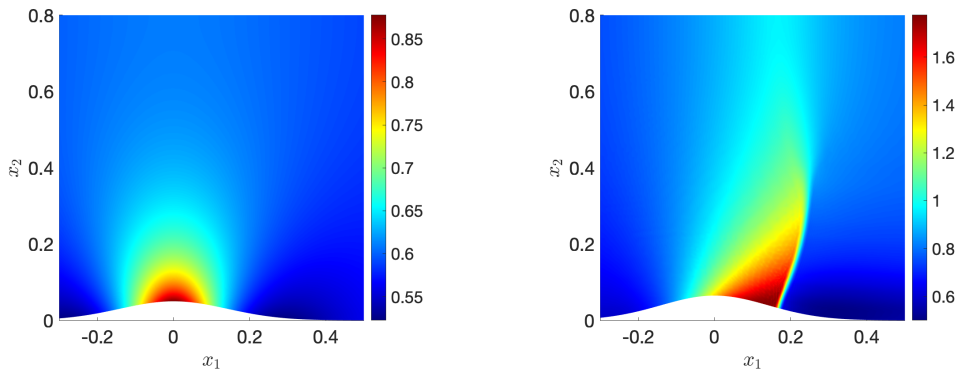
completed with wall boundary conditions on top and bottom boundaries, subsonic inlet condition (total temperature, total pressure and flow direction) at the left boundary and subsonic outlet condition (static pressure) at the right boundary — the symbol $(\cdot)^\top$ denotes the transposition operator. We express the free-stream field q_∞ in terms of the Mach number Ma_∞,

$$T_\infty = 1, \quad p_\infty = \frac{1}{\gamma}, \quad \rho_\infty = 1, \quad u_\infty = \mathrm{Ma}_\infty e_1.$$

Finally, we introduce the parameter vector $\mu = [h, \mathrm{Ma}_\infty]$ and the parameter region $\mathcal{P} = [0.05, 0.065] \times [0.58, 0.78]$. Note that the computational domain Ω depends on the geometric parameter h; therefore, we should introduce a geometric mapping to recast the problem over a parameter-independent configuration. We here resort to a Gordon-Hall transformation; we refer to [20, section 2] for the details.



(a)



(b) (c)

Figure 2: inviscid flow over a Gaussian bump. (a) computational domain. (b)-(c) visualization of the Mach field for $\mu = [0.05, 0.58]$ and $\mu = [0.065, 0.78]$.

Figures 2(b) and (c) show the behavior of the Mach field for two values of the parameters in $\mathcal{P}$: we observe that the flow is completely subsonic for moderate values of Ma_∞ and develops a normal shock over the bump for $\mathrm{Ma}_\infty \gtrsim 0.65 - 0.7$.

3 Parametric mesh adaptation

We consider the problem of determining an adapted mesh $\mathcal{T}_{\mathrm{hf}}$ of the domain Ω based on a set of snapshots $\widetilde{\mathcal{S}}_{\mathrm{hf}} = \{\widetilde{q}_\mu^{\mathrm{hf}} : \mu \in \mathcal{P}_{\mathrm{train}}\}$ defined over a mesh $\mathcal{T}_{\mathrm{hf}}^0$. We here pursue a metric-based MA approach: first, we resort to the snapshots in $\widetilde{\mathcal{S}}_{\mathrm{hf}}$ to define a metric $\mathfrak{M}$; then, we resort to a MA toolkit to devise the P1 mesh and subsequently we use standard FE routines to devise the high-order mesh. In section 3.1, we review the key elements of anisotropic mesh adaptation; then, in section 3.2 we briefly comment on mesh adaptation for one-dimensional problems; in section 3.3, we discuss an isotropic mark-then-refine strategy employed for the two-dimensional problem of section 2.2.2; finally, in section 3.4, we review Hessian-based anisotropic mesh adaptation, and we discuss metric intersection for parametric problems. We refer to [29, 30] for a thorough introduction to metric-based MA.

3.1 Fundamentals of anisotropic metric-based mesh adaptation

A Riemannian metric field is derived from the error estimate and prescribes the size and shape of the mesh elements.

Euclidean space. A scalar product is a symmetric positive definite (SDP) form, which can be represented by an SPD matrix $\mathfrak{M}$, which is dubbed *metric tensor* or simply *metric*. The scalar product is then written:

$$(x,\, y)_{\mathfrak{M}} \;=\; x^\top \mathfrak{M}\, y \quad \text{where} \quad x, y \in \mathbb{R}^d. \tag{14}$$

A vector space with a scalar product is called an *Euclidean space*. The scalar product is associated with a distance that can be used to compute lengths in the Euclidean space :

$$\ell_{\mathfrak{M}}(x,y) = \sqrt{(y-x)^\top \mathfrak{M}\,(y-x)}\,, \tag{15}$$

from which we deduce classic geometrical quantities such as angles or volumes.

The metric $\mathfrak{M}$ is diagonalizable in an orthonormal basis:

$$\mathfrak{M} = \mathcal{R}^\top \Lambda\, \mathcal{R}\,, \tag{16}$$

where $\Lambda = \mathrm{diag}\,(\lambda_1,\, \ldots,\, \lambda_d)$ is the diagonal matrix of eigenvalues of $\mathfrak{M}$ and $\mathcal{R} = (\,\mathbf{r}_1 \,|\, \mathbf{r}_2 \,|\, \ldots \,|\mathbf{r}_d\,)$ is the unitary matrix (*i.e.* $\mathcal{R}^\top \mathcal{R} = \mathcal{I}_d$) of eigenvectors of $\mathfrak{M}$. A metric tensor has an intuitive geometric representation: the set of points that are at constant distance from a point $x \in \mathbb{R}^d$ is an ellipsoid centered in x whose axes are aligned with the eigenvectors $\mathbf{r}_1, \ldots, \mathbf{r}_d$ of $\mathfrak{M}$. The set of points at distance one from a point (the unit ball of $\mathfrak{M}$), is an ellipsoid for which the sizes of the axes are $h_i = \lambda_i^{-\frac{1}{2}}$. In other words, in the context of anisotropic mesh adaptation, the eigenvectors of the metric tensor prescribe the orientation of the elements, while the eigenvalues prescribe the sizes in these directions. In an Euclidean metric space, these sizes and orientations are constant over all the domain; for mesh adaptation, we want them to vary in space depending on the solution features. This observation motivates the introduction of Riemannian metric spaces.

Riemannian metric space. We now define a metric tensor field $\mathfrak{M} : \Omega \to \mathbb{R}^{d\times d}$ such that $\mathfrak{M}(x)$ is symmetric positive definite for all $x \in \Omega$. There is no notion of global scalar product; however, we can extend the notion of distance. Given $x, y \in \Omega$, we define the distance

$$\ell_{\mathfrak{M}}(x,y) = \int_0^1 \sqrt{(y-x)^\top \mathfrak{M}((1-t)x+ty)(y-x)}\, dt\,, \tag{17}$$

and, given a set $A \subset \Omega$, we define the volume:

$$\big|A\big|_{\mathfrak{M}} = \int_A \sqrt{\det(\mathfrak{M}(x))}\, dx. \tag{18}$$

Locally, the eigenvalues and the eigenvectors of the metric tensor $\mathfrak{M}(x)$ define size and orientations, respectively.

Unit mesh. Given the mesh $\mathcal{T}_{\text{hf}}$ of Ω with elements $\{\mathtt{D}_k\}_{k=1}^{N_\text{e}}$, we say that an element $\mathtt{D}_k$ is a quasi-unit element with respect to $\mathfrak{M}$ if the lengths (17) of all its edges are approximately equal to one and its volume is approximately equal to $\frac{\sqrt{3}}{4}$ for $d=2$ and $\frac{\sqrt{2}}{12}$ for $d=3$. Similarly, we say that the mesh $\mathcal{T}_{\text{hf}}$ is unit if all its elements are quasi-unit. Adapting a mesh with respect to $\mathfrak{M}$ comes to generating a mesh that is unit in that metric field.

Remark 3.1. *Exploiting the geometric interpretation of the metric tensor, we can devise a practical strategy to identify the metric tensor $\mathfrak{M}$ associated to the triangle $\mathtt{D}$: we shall use this strategy in section 3.3. We define the vertices $\{x_1^\text{v}, x_2^\text{v}, x_3^\text{v}\}$ so that the longest (in Euclidean norm) edge is the one that connects x_1^v and x_2^v. First, we set λ_1 equal to the square of the inverse of $\|x_2^\text{v} - x_1^\text{v}\|_2$, and λ_2 equal to the square of the inverse of the distance between the vertex x_3^v and the edge $\overline{x_1^\text{v} x_2^\text{v}}$; second, we set $\mathbf{n}_1 = \frac{x_2^\text{v} - x_1^\text{v}}{\|x_2^\text{v} - x_1^\text{v}\|_2}$ and $\mathbf{n}_2 = [(\mathbf{n}_1)_2, -(\mathbf{n}_1)_1]^\top$; finally, we define $\mathcal{R} = [\mathbf{n}_1, \mathbf{n}_2]$, the diagonal matrix $\Lambda = \operatorname{diag}(\lambda_1, \lambda_2)$ and the metric $\mathfrak{M} = \mathcal{R}^\top \Lambda \mathcal{R}$.*

3.2 Mesh adaptation for one-dimensional problems

For one-dimensional problems, we resort to the standard de Boor's algorithm (see, e.g., [22, Chapter 2]): for consistency with section 3.1, we present the method in a slightly different formalism than the one of [22]. Given the metric $\mathfrak{M} : \Omega \to \mathbb{R}^+$, we define the mesh density function $\mathfrak{d} : \Omega \to \mathbb{R}^+$:

$$\mathfrak{d}(x) = \sqrt{\mathfrak{M}(x)}\,, \quad x \in \Omega. \tag{19}$$

Our goal is to construct a (quasi-)unit mesh with N nodes $\{x_i^\text{hf}\}_{i=1}^N$ with respect to the metric $\mathfrak{M}$, that is

$$\int_{x_1^\text{hf}}^{x_N^\text{hf}} \mathfrak{d}(x)\,dx = N \quad \text{and} \quad \int_{x_i^\text{hf}}^{x_{i+1}^\text{hf}} \mathfrak{d}(x)\,dx = 1\,, \quad i = 1, \dots, N-1\,. \tag{20}$$

De Boor's algorithm constructs a unit mesh for an approximate metric. First, we introduce an initial grid $x_1^{\text{hf},0} \le \ldots \le x_{N_0}^{\text{hf},0}$ and the piecewise-constant approximation $\widehat{\mathfrak{d}}$ of $\mathfrak{d}$ such that,

$$\widehat{\mathfrak{d}}(x) = \frac{1}{x_{i+1}^{\text{hf},0} - x_i^{\text{hf},0}} \int_{x_i^{\text{hf},0}}^{x_{i+1}^{\text{hf},0}} \mathfrak{d}(x)\,dx, \quad x \in \left(x_i^{\text{hf},0}, x_{i+1}^{\text{hf},0}\right), \;\; i = 1, \dots, N-1.$$

Then, we find the unique set of points $\{x_i\}_{i=1}^N$ that satisfies (20) for the mesh density function $\widehat{\mathfrak{d}}$: since $\widehat{\mathfrak{d}}$ is piecewise-constant, we can obtain an explicit expression for $\{x_i\}_{i=1}^N$. We refer to [22] for the explicit formula. Note that several iterations of this algorithm are typically required to obtain a good approximation of the equidistributed mesh for $\mathfrak{d}$.

For the nozzle flow problem, we define the mesh density function $\mathfrak{d}$ based on the second-order derivative of the Mach number. Given the mapped fields $\{\widetilde{q}_\mu^\text{hf} : \mu \in \mathcal{P}_\text{train}\}$, we define the corresponding Mach fields $\{\widetilde{\text{Ma}}_\mu^\text{hf} : \mu \in \mathcal{P}_\text{train}\}$ and the non-normalized density ρ

$$\rho(x) = \max_{\mu \in \mathcal{P}_\text{train}} \max\left\{\left|\partial_{xx}\widetilde{\text{Ma}}_\mu^\text{hf}(x)\right|, C_\mu\right\} \quad \text{where} \quad C_\mu = 10^{-2} \sup_{x \in \Omega} \left|\partial_{xx}\widetilde{\text{Ma}}_\mu^\text{hf}(x)\right|. \tag{21}$$

Finally, we define the normalized density:

$$\mathfrak{d}(x) = \frac{N}{\int_\Omega \rho(x)\,dx}\rho(x).$$

The function ρ (and thus $\mathfrak{d}$) is well-defined in the interior of each element of the mesh $\mathcal{T}_\text{hf}^0$; in our implementation, we rely on the evaluation of the sensor (21) in the elements' quadrature points to define the piecewise-constant function that is used by the de Boor's algorithm.

3.3 Isotropic mark-then-refine mesh adaptation

For the Euler equations, the total enthalpy H_tot (cf. (1)) is constant and can be computed exactly from the boundary conditions for any $\mu \in \mathcal{P}$. Given the set of snapshots $\widetilde{\mathcal{S}}_\text{hf}$ defined over the mesh $\mathcal{T}_\text{hf}^0$, we first compute the average error in total enthalpy

$$\eta_{\mu,k} = \frac{1}{|\mathtt{D}_k^0|} \int_{\mathtt{D}_k^0} \left(H_{\text{tot},\mu}^\text{hf} - H_{\text{tot},\mu}^\text{true}\right)^2 dx, \quad k = 1, \dots, N_\text{e}^0, \tag{22a}$$

with $H^{\text{true}}_{\text{tot},\mu} := H_{\text{tot}}(q^{\text{true}}_\mu)$ and $H^{\text{hf}}_{\text{tot},\mu} := H_{\text{tot}}(q^{\text{hf}}_\mu)$, and the maximum over the training set of parameters

$$\eta^{\max}_k = \max_{\mu\in\mathcal{P}_{\text{train}}} \eta_{\mu,k}, \quad k = 1,\dots,N^0_{\text{e}}. \tag{22b}$$

Second, given $\gamma_{\text{ref}} \in (0,1)$, we mark the $\gamma_{\text{ref}} \cdot N^0_{\text{e}}$ elements that maximize $\{\eta^{\max}_k\}_k$; we denote by $\mathtt{I}_{\max} \subset \{1,\dots,N^0_{\text{e}}\}$ the indices of the marked elements. Third, we extract the metric $\{\mathfrak{M}^0_k\}_{k=1}^{N^0_{\text{e}}}$ from the mesh $\mathcal{T}^0_{\text{hf}}$ using the strategy in Remark 3.1 and we define the new metric as follows:

$$\mathfrak{M}_k = \mathfrak{M}^0_k \text{ if } k \notin \mathtt{I}_{\max}, \quad \mathfrak{M}_k = 4\mathfrak{M}^0_k \text{ if } k \in \mathtt{I}_{\max}. \tag{23}$$

Multiplication by four in (23) corresponds to an isotropic reduction of the local mesh density by a factor of two. Fourth, we define the metric $\mathfrak{M}$ in the vertices of the mesh $\mathcal{T}^0_{\text{hf}}$ using the simple average:

$$\mathfrak{M}_i = \frac{1}{\#\text{Neigh}_i} \sum_{k\in\text{Neigh}_i} \mathfrak{M}_k, \quad i = 1,\dots,N^0_{\text{v}},$$

where $\text{Neigh}_i \subset \{1,\dots,N^0_{\text{e}}\}$ contains the indices of the elements that contain the i-th vertex of $\mathcal{T}^0_{\text{hf}}$. Finally, we apply a mesh adaptation toolkit to generate the new mesh.

The choice of γ_{ref} regulates how quickly we increase the size of the HF mesh. Since the mesh $\mathcal{T}_{\text{hf}}$ should be accurate for the mapped manifold $\{q^{\text{true}}_\mu \circ \Phi_\mu : \mu \in \mathcal{P}\}$ where the mapping Φ changes at each outer-loop iteration, it is not worth to refine the mesh if the mapping Φ is excessively inaccurate. On the other hand, modest values of γ_{ref} might require a large number of iterations in Algorithm 1. A thorough investigation of the choice of γ_{ref} on performance is beyond the scope of the present work; in the numerical experiments, we mark 10% of the elements (i.e., $\gamma_{\text{ref}} = 10\%$) of the mesh at each iteration.

We observe that the computation of (22) for all $\mu \in \mathcal{P}_{\text{train}}$ might be expensive: in the numerical experiments, we hence run a strong greedy algorithm (cf. Appendix A) to identify the most relevant parameters in $\mathcal{P}_{\text{train}}$. We also notice that the choice of the error indicator (22) is specific to the Euler equations: several alternatives have been considered in the literature such that the $\mathtt{p}+1$ residual or goal-oriented error estimates (e.g., [55]). Finally, we observe that at each iteration of Algorithm 1 we generate a new mesh that is independent of the previous meshes: the advantage of this choice is that we allow the mesh adaptation toolkit to perform smoothing operations that ensure well-behaved meshes; clearly, the disadvantage is that we need to perform mesh interpolation between unstructured meshes at several steps of our training phase.

3.4 Anisotropic mesh adaptation

Anisotropic mesh adaptation refers to a class of methods where both the elements sizes and orientations are optimized with respect to an error estimate. As explained in section 3.1, this can be achieved by generating a unit mesh with respect to a prescribed metric field. In this work, we use the multiscale metric defined in [30]:

$$\mathfrak{M}_s(x) = \left(\frac{N}{\int_\Omega (\det(|H_s(\bar{x})|)\mathrm{d}\bar{x})^{\frac{p}{2p+d}}}\right)^{2/d} \det(|H_s(x)|)^{\frac{-1}{(2p+d)}} |H_s(x)|, \quad \forall x \in \Omega\,, \tag{24}$$

where $s : \Omega \to \mathbb{R}$ is a scalar field driving the adaptation, $H_s : \Omega \to \mathbb{R}^{d\times d}$ is the Hessian matrix of s. Since H_s is real-valued and symmetric, it can be diagonalized: $H_s = \mathcal{R}_s^\top \Lambda_s \mathcal{R}_s$ where $\mathcal{R}_s$ is the orthogonal matrix of eigenvectors and $\Lambda_s = \text{diag}(\lambda_{s,1}, \lambda_{s,2})$ is the diagonal matrix of eigenvalues. We further define $|H_s|$ by taking the absolute value of the eigenvalues: $|H_s| := \mathcal{R}_s^\top \text{diag}(|\lambda_{s,1}|, |\lambda_{s,2}|)\mathcal{R}_s$. The constant p is associated with the L^p norm that is used to derive the error estimate and ultimately the metric. On the other hand, N is the *continuous complexity* and is directly related to the target number of vertices of the mesh.

In this work, we do not only want to adapt to one field, but to as many fields as we have snapshots. Instead of generating several meshes and taking the supermesh of several adapted meshes, we choose to construct one metric based on the available fields and generate only one adapted mesh. A fairly common strategy is used to construct that unique metric: the multiscale metric (from Eq. (24)) is computed for each snapshot, and those metrics are then combined vertex-wise using metric intersection :

$$\mathfrak{M}(x) = \bigcap_{k=1}^{n_{\text{train}}} \mathfrak{M}_{s^k}(x)\,, \tag{25}$$

where n_{train} is the number of snapshots in $\mathcal{P}_{\text{train}}$ and $\mathfrak{M}_{s^k}$ is the multiscale metric field computed for the chosen scalar field of the k-th snapshot of the dataset.

A procedure to intersect two metrics is proposed in [4]. Geometrically, we have seen that the unit ball of a metric is an ellipsoid. Intersecting two metrics is equivalent to finding the largest ellipsoid included in the two

corresponding ellipsoids. Note that this method has limitations, which we aim to solve in future work. First, metric intersection is not associative, which means that depending on the order in which the intersections are carried out, the final result will slightly change, although it is not known to change significantly. Second, if each metric field associated with each snapshot would result in a mesh with a certain prescribed number of vertices, once the metrics are intersected, we do not control the new prescribed number of vertices. To avoid this issue, we could intersect the Hessians and then apply the multiscale normalization. However, the geometric interpretation of intersecting quantities with potentially very different orders of magnitudes is unclear.

3.5 Practical considerations

Good approximations of the gradient and the Hessian of the solution field are key in the implementation of the metric. If the order of the solver is high enough (strictly greater than one), those quantities are computed exactly in each element. Otherwise, we perform L^2 projections based on Clement's interpolation [14] to reconstruct a P^1 gradient (resp. Hessian) from a P^0 gradient (resp. Hessian). Metric (24) depends on the choice of scalar field s, and parameter p. We here use the Mach number field as sensor to drive the adaptation. In the numerical simulations, we set $p = 1$, which is found to better capture small-scale features of the solution field.

The task of generating the adapted meshes is left to the anisotropic remesher `mmg2d` ([2, 16]). It takes as input a mesh and a metric field defined on the mesh, and returns a unit mesh for the given metric. To do so, it performs iteratively a series of local mesh operations : vertex addition, removal, smoothing and topology changes. In this work, the default parameters of `mmg2d` are used. The eigenvalues of metric (24) are previously truncated to avoid excessively small edge sizes.

Mesh adaptation is a non-linear problem, where the convergence of the mesh/solution couple has to be considered: a better mesh gives a better solution, which in turns gives a better mesh, etc. This is addressed in Algorithm 1: during each outer loop iteration, we indeed generate snapshots (cf. Line 3) associated to the current mesh and we use them to adapt the mesh (cf. Line 4). We hence expect to converge to a final mesh — and a reduced-order approximation — that is accurate for all parameters in $\mathcal{P}$.

4 Registration

The second ingredient of our method is a registration algorithm that is designed to track coherent structures of the solution field, to facilitate the tasks of mesh adaptation and linear-subspace model reduction; the algorithm takes as input (i) a set of snapshots $\{q_\mu^{\mathrm{hf}} : \mu \in \mathcal{P}_{\mathrm{train}}\}$ and (ii) a mesh $\mathcal{T}_{\mathrm{hf}}$ of the domain Ω, and returns a parameterized map $\Phi : \Omega \times \mathcal{P} \to \Omega$ such that $\Phi_\mu(\mathcal{T}_{\mathrm{hf}})$ is a proper mesh of Ω for all $\mu \in \mathcal{P}$. The development and the analysis of registration methods for MOR remains a challenging task that requires many advances; in this work, we briefly summarize the procedure employed in the numerical experiments and we refer to a future work for a thorough discussion on registration methods.

4.1 Spectral maps

Given the family of domains $\{\Omega_\mu : \mu \in \mathcal{P}\}$, we define the "reference" domain Ω_{p} and the geometric map $\Psi^{\mathrm{geo}} : \Omega_{\mathrm{p}} \times \mathcal{P} \to \mathbb{R}^d$ such that $\Psi_\mu^{\mathrm{geo}}(\Omega_{\mathrm{p}}) = \Omega_\mu$ for all $\mu \in \mathcal{P}$. We denote by $\mathbf{n}_{\mathrm{p}}$ the outward normal to $\partial\Omega_{\mathrm{p}}$ and we define the space of tensorized polynomials $\mathbb{Q}_J$ of degree at most J in each variable. For the nozzle problem, we consider $\Psi^{\mathrm{geo}} = \mathtt{id}$ — where $\mathtt{id}(x) = x$ is the identity map — and $\Omega_{\mathrm{p}} = \Omega$; for the transonic bump problem, we consider a Gordon-Hall map (cf. [20, section 2]) and $\Omega_{\mathrm{p}} = (0, 1)^2$; in the latter, we introduce the reference parameter $\bar{\mu} \in \mathcal{P}$ and we define $\Omega := \Omega_{\bar{\mu}}$. Then, we consider mappings of the form

$$\mathtt{N}_\mu(\mathbf{a}) \;=\; \Psi_\mu^{\mathrm{geo}} \circ \mathtt{N}_{\mathrm{p}}(\mathbf{a}) \circ \Lambda_{\bar{\mu}}^{\mathrm{geo}}, \quad \mathtt{N}_{\mathrm{p}}(\mathbf{a}) = \mathtt{id} + \sum_{i=1}^{m} (\mathbf{a})_i \varphi_i, \tag{26a}$$

where $\Lambda_{\bar{\mu}}^{\mathrm{geo}} := \left(\Psi_{\bar{\mu}}^{\mathrm{geo}}\right)^{-1}$, $\{\varphi_i\}_{i=1}^m$ spans the space $\mathcal{U}_{\mathrm{p}}$ of tensorized polynomials such that

$$\mathcal{U}_{\mathrm{p}} = \mathrm{span}\{\varphi_i\}_{i=1}^m \subset \mathcal{U}_{\mathrm{hf,p}} = \left\{\varphi \in [\mathbb{Q}_J]^d : \; \varphi \cdot \mathbf{n}_{\mathrm{p}}|_{\partial\Omega_{\mathrm{p}}} = 0\right\}. \tag{26b}$$

We equip $\mathcal{U}_{\mathrm{hf,p}}$ with the inner product

$$(\varphi, \psi)^2_{H^2(\Omega_{\mathrm{p}})} := \int_{\Omega_{\mathrm{p}}} \left(\sum_{i,j,k=1}^{d} \partial_{j,k}(\varphi)_i \cdot \partial_{j,k}(\psi)_i \;+\; \varphi \cdot \psi \right) dx, \tag{27}$$

and we assume that $\{\varphi_i\}_{i=1}^m$ is an orthonormal basis of $\mathcal{U}_{\mathrm{p}}$. We observe that, if $\mathtt{N}_{\mathrm{p}}(\mathbf{a})$ is a bijection from Ω_{p} in itself, $\mathtt{N}_\mu(\mathbf{a})$ is a bijection from Ω to Ω_μ, for all $\mu \in \mathcal{P}$.

We denote by $\mathcal{B}_\mu$ the space of diffeomorphisms from Ω to Ω_μ; exploiting the analysis in [43, 47], we can prove that (i) for any $\mu \in \mathcal{P}$ the model class $\mathtt{N}$ (26) is dense in a meaningful subspace of $\mathcal{B}_\mu$ and (ii) the set of admissible maps $A_{\mathrm{bj},\mu} = \{\mathbf{a} \in \mathbb{R}^m : \mathtt{N}_\mu(\mathbf{a}) \in \mathcal{B}_\mu\}$ has a non-empty interior for any choice of $\mathcal{U}_\mathrm{p}$ in (26b). The latter is extremely important for the numerical robustness of registration methods. We remark that the two results are currently restricted to domains that are diffeomorphic to the unit hyper-cube $\Omega_\mathrm{p} = (0,1)^d$. The extension of these results to a broader class of domains is the subject of ongoing research.

4.2 Optimization-based registration

Given the training set of snapshots $\{q_\mu^\mathrm{hf} : \mu \in \mathcal{P}_\mathrm{train}\}$, we determine the mapping coefficients $\widehat{\mathbf{a}}_\mu$ (cf. (3)) by solving the optimization problem:

$$\min_{\mathbf{a}\in\mathbb{R}^m} \mathfrak{f}_\mu^\mathrm{obj}(\mathbf{a}) := \mathfrak{f}_\mu^\mathrm{tg}(\mathbf{a}) \,+\, \xi \left(\left|\mathtt{N}_\mathrm{p}(\mathbf{a})\right|^2_{H^2(\Omega_\mathrm{p})} \,+\, \mathfrak{f}_\mathrm{msh}(\mathtt{N}_\mu(\mathbf{a})) \,+\, \mathfrak{f}_\mathrm{jac}(\mathtt{N}_\mathrm{p}(\mathbf{a})) \right). \tag{28}$$

Here, $\mathfrak{f}_\mu^\mathrm{tg}$ denotes the target (or proximity) function that measures the degree of similarity between the available estimate of the solution field q_μ^true and a suitable template solution or template reduced space, while the terms multiplied by the weighting parameter $\xi > 0$ are regularization terms that promote the smoothness of the map and ensure bijectivity. In more detail, $\left|\cdot\right|^2_{H^2(\Omega_\mathrm{p})} = \int_{\Omega_\mathrm{p}} \left(\sum_{i,j,k=1}^d \partial_{j,k}(\cdot)_i^2 \right) dx$ is the H^2 seminorm; $\mathfrak{f}_\mathrm{msh}$ controls the quality of the deformed mesh (cf. [58]),

$$\mathfrak{f}_\mathrm{msh}(\Phi) = \frac{1}{|\Omega|} \sum_{k=1}^{N_\mathrm{e}^\mathrm{pb}} |\mathtt{D}_k^\mathrm{pb}| \int_{\widehat{\mathtt{D}}} \exp\left(\frac{q_k^\mathrm{msh}(\Phi)}{q_k^\mathrm{msh}(\mathtt{id})} \,-\, \kappa_\mathrm{msh} \right) dx, \quad q_k^\mathrm{msh}(\Phi) := \frac{1}{d^2} \left(\frac{\|\nabla \Psi^\mathrm{hf}_{\Phi,k}\|_\mathrm{F}^2}{(\det(\nabla \Psi^\mathrm{hf}_{\Phi,k}))_+^{2/d}} \right)^2, \tag{29}$$

where $\{\Psi^\mathrm{hf}_{\Phi,k}\}_k$ are the elemental maps (6) associated to the deformed mesh $\Phi(\mathcal{T}_\mathrm{hf})$; and $\mathfrak{f}_\mathrm{jac}$ is designed to ensure that the selected map is non-singular,

$$\mathfrak{f}_\mathrm{jac}(\Phi_\mathrm{p}) = \frac{1}{|\Omega_\mathrm{p}|} \int_{\Omega_\mathrm{p}} \exp\left(\frac{\epsilon - \det(\nabla \Phi_\mathrm{p})}{C_\mathrm{exp}} \right) dx, \quad \text{with } \epsilon \in (0,1),\ C_\mathrm{exp} \ll \epsilon. \tag{30}$$

Note that (29) and (30) depend on several hyper-parameters: in the numerical experiments, we consider

$$\epsilon = 0.1, \quad C_\mathrm{exp} = 0.025\epsilon, \quad \kappa_\mathrm{msh} = 10, \quad \xi = 10^{-3}.$$

We observe that $q_k^\mathrm{msh}(\Phi) \equiv 1$ for $d = 1$ dimensional problems that are discretized using linear elements: we hence omit the mesh regularization term for the nozzle problem. Furthermore, we empirically found that the regularization (30) is not strictly needed for two-dimensional problems based on discretize-then-map treatment of geometry parameterizations (cf. section 5): in the numerical experiments we hence omit the regularization (30) for the transonic bump test case. In the remainder of this section, we discuss the choice of the target function for the two model problems considered in the numerical section.

Target function for the nozzle problem

Given the snapshot q_μ^hf, we compute the Mach field $\mathrm{Ma}_\mu^\mathrm{hf}$ and we estimate the maximum of its derivative $x_\mu^\star$; then, we consider the target

$$\mathfrak{f}_\mu^\mathrm{tg}(\mathbf{a}) = \left|\mathtt{N}(x_{\bar{\mu}}^\star; \mathbf{a}) \,-\, x_\mu^\star\right|^2. \tag{31}$$

In the numerical experiments, we estimate $x_\mu^\star$ using the formula

$$x_\mu^\star = \frac{1}{\#\mathtt{I}_+} \sum_{i\in\mathtt{I}_+} x_i^{\mathrm{hf,qd}}, \quad \mathtt{I}_+ = \left\{ i \in \{1,\dots,N_\mathrm{hf,q}\} : |\partial_x \mathrm{Ma}_\mu^\mathrm{hf}(x_i^{\mathrm{hf,qd}})| > \delta \max_j |\partial_x \mathrm{Ma}_\mu^\mathrm{hf}(x_i^{\mathrm{hf,qd}})| \right\}, \tag{32}$$

where $\{x_i^{\mathrm{hf,qd}}\}_{i=1}^{N_\mathrm{hf,q}}$ are the quadrature points of the FE mesh and $\delta > 0$ is a threshold that is set equal to 0.5. We observe that the definitions of (31) and (32) exploit the knowledge that the solution exhibits a single discontinuity in Ω; we refer to [24] and [44] for a generalization to a more general setting.

Target function for the transonic bump problem

We consider the target

$$\mathfrak{f}_\mu^\mathrm{tg}(\mathbf{a}) = \min_{\nu\in\mathcal{S}_n} \frac{1}{|\Omega_\mathrm{p}|} \int_{\Omega_\mathrm{p}} \left| s_\mu^\mathrm{hf} \circ \mathtt{N}_\mathrm{p}(\mathbf{a}) - \nu \right|^2 \det(\nabla \Psi_{\bar{\mu}}) dx \,+\, \|\mathtt{N}_\mu(x_{\bar{\mu}}^\star; \mathbf{a}) \,-\, x_\mu^\star\|_2^2, \tag{33}$$

where $s^{\rm hf}_\mu = {\rm Ma}^{\rm hf}_\mu \circ \Psi^{\rm geo}_\mu$, and $x^\star_\mu$ is equal to the maximum of the Mach number over the bump if the flow is subsonic, and equal to the maximum of the tangential derivative of the Mach number — which is practically estimated using (32) — if the flow is transonic. We observe that the evaluation of the target (33) requires the evaluation of the field $s^{\rm hf}_\mu$ in arbitrary points of $\Omega_{\rm p}$: it is thus important to define $s^{\rm hf}_\mu$ over a structured grid.

For completeness, we comment on the choice of the first term in (33). Given the reduced space $\widetilde{\mathcal{S}}_n \subset L^2(\Omega_\mu)$, the goal of registration is to find a mapping Φ such that

$$\min_{\nu \in \widetilde{\mathcal{S}}_n} \int_{\Omega_\mu} \left| {\rm Ma}^{\rm hf}_\mu \circ \Phi - \nu \right|^2 dx,$$

where the choice to consider the Mach number as registration sensor is justified by the observation that it is a scalar quantity that exhibits relevant features (shocks, contact discontinuities) of the full field $q^{\rm true}_\mu$. Exploiting the expression of Φ, $\Phi = \Psi^{\rm geo}_\mu \circ \Phi_{\rm p} \circ \Lambda^{\rm geo}_{\bar\mu}$, and the change of variable $x = \Psi^{\rm geo}_{\bar\mu}(\xi)$, we find

$$\min_{\nu \in \widetilde{\mathcal{S}}_n} \int_{\Omega_\mu} \left| s^{\rm hf}_\mu \circ \Phi - \nu \right|^2 \det(\nabla \Psi_{\bar\mu}) dx \quad \text{where } \mathcal{S}_n = \left\{ \nu \circ \Psi^{\rm geo}_\mu \,:\, \nu \in \widetilde{\mathcal{S}}_n \right\}, \quad s^{\rm hf}_\mu = {\rm Ma}^{\rm hf}_\mu \circ \Psi^{\rm geo}_\mu.$$

The space $\mathcal{S}_n \subset L^2(\Omega_{\rm p})$ in (33) is dubbed *template space* and is built using the greedy procedure proposed in [48].

4.3 Parametric registration

We combine the optimization statement discussed in the previous section with the greedy algorithm proposed in [48] for the adaptive construction of the template space $\mathcal{S}_n$ in (33), and a standard regression procedure to obtain the parametric mapping Φ — for completeness, we report the greedy method in Appendix A. For the nozzle problem, the greedy procedure is not necessary: in this case we simply rely on [43, Algorithm 1]. In both cases, the cost of the procedure is dominated by the solution to the optimization statement (28) for all $\mu \in \mathcal{P}_{\rm train}$,

$$\widehat{\mathbf{a}}_\mu \in \arg\min_{\mathbf{a} \in \mathbb{R}^m} \mathfrak{f}^{\rm obj}_\mu(\mathbf{a}), \quad \mu \in \mathcal{P}_{\rm train} \tag{34}$$

for the first iteration of the algorithm — which corresponds to the choice $\mathcal{S}_1 = {\rm span}\{s^{\rm hf}_{\bar\mu}\}$ for the transonic bump test case.

We rely on the Matlab function `fminunc` which implements a quasi-Newton method; since the problem is non-convex, the choice of the initial condition for the optimizer is critical to achieve accurate performance. Towards this end, following [43], we first reorder the parameters in $\mathcal{P}_{\rm train}$ so that $\mu^{(1)} = \arg\min_{\mu \in \mathcal{P}_{\rm train}} \|\mu - \bar\mu\|_2$ and

$$\mu^{(k)} = \arg \min_{\mu \in \mathcal{P}_{\rm train} \setminus \{\mu^{(i)}\}_{i=1}^{k-1}} \left(\min_{\mu' \in \{\mu^{(i)}\}_{i=1}^{k-1}} \|\mu - \mu'\|_2 \right), \quad k = 2, \ldots, n_{\rm train};$$

then, we choose the initial condition as follows:

$$\mathbf{a}^0_{\mu^{(1)}} = 0, \quad \mathbf{a}^0_{\mu^{(k)}} = \widehat{\mathbf{a}}_{\mu^{({\rm ne}_k)}}, \;\; \text{with } {\rm ne}_k = \arg\min_{j=1,\ldots,k-1} \|\mu^{(j)} - \mu^{(k)}\|_2, \quad k = 2, \ldots, n_{\rm train}.$$

We observe that this choice of the initial condition prevents the parallelization of the registration procedure.

Remark 4.1. *In the numerical experiments for the two-dimensional test case, we consider polynomials of degree $J = 10$ and we rely on a P1 61×21 Cartesian FE grid of the unit square to represent the sensors $\mu \mapsto s^{\rm hf}_\mu$. If we denote by $\{x^{\rm hf,r}_j\}_{j=1}^{N_{\rm nd,r}}$ the nodes of the mesh on $\Omega_{\rm p}$, computation of $s^{\rm hf}_\mu$ requires the interpolation of the FE field ${\rm Ma}^{\rm hf}_\mu$ in the points $\{\Psi^{\rm geo}_\mu(x^{\rm hf,r}_j)\}_{j=1}^{N_{\rm nd,r}}$. To ensure that the objective function is sufficiently smooth for gradient-based optimization, we post-treat the sensor by applying a low-pass filter (moving average) in each spatial direction.*

5 Linear-subspace projection-based model order reduction

In this section, we present the projection-based MOR procedure employed to estimate the mapped field $\widetilde{q}^{\rm true}_\mu := q^{\rm true}_\mu \circ \Phi_\mu$. As anticipated in the introduction, we seek approximations of the form

$$\widetilde{q}_\mu = \mathtt{Z}\widehat{\boldsymbol{\alpha}}_\mu \quad \text{with } \widehat{\boldsymbol{\alpha}}_\mu \in \arg\min_{\boldsymbol{\alpha} \in \mathbb{R}^n} \max_{\psi \in \widehat{\mathcal{Y}}} \frac{\mathfrak{R}^{\rm eq}_\mu(\mathtt{Z}\boldsymbol{\alpha}, \psi)}{|||\psi|||} \tag{35a}$$

where $\mathtt{Z} : \mathbb{R}^n \to \mathcal{X}_{\rm hf}$ is a suitable linear operator,

$$\mathfrak{R}^{\rm eq}_\mu(q,v) = \sum_{k=1}^{N_{\rm e}} \rho^{{\rm eq,e}}_k \, r^{\rm e}_{k,\mu}(E_k q, E_k v) \; + \; \sum_{j=1}^{N_{\rm f}} \rho^{{\rm eq,f}}_k r^{\rm f}_{j,\mu}(E_j^+ q, E_j^- q, E_j^+ v, E_j^- v), \quad \forall\, q, v \in \mathcal{X}_{\rm hf}, \tag{35b}$$

is a weighted residual that depends on the sparse weights $\boldsymbol{\rho}^{\rm eq,e} \in \mathbb{R}^{N_{\rm e}}$ and $\boldsymbol{\rho}^{\rm eq,f} \in \mathbb{R}^{N_{\rm f}}$, $\widehat{\mathcal{Y}} \subset \mathcal{X}_{\rm hf}$ is a m-dimensional linear space with $m \geq n$, and $|||\cdot||| = \sqrt{((\cdot,\cdot))}$ is the norm associated to the test space. As in [20], we consider a discrete L^2 inner product for the trial space and a discrete H^1 inner product for the test space such that

$$\begin{cases} (q,v) = \displaystyle\sum_{k=1}^{N_{\rm e}} \int_{\mathtt{D}_k} q \cdot v \, dx \\ ((q,v)) \; = \displaystyle\sum_{k=1}^{N_{\rm e}} \int_{\mathtt{D}_k} (\nabla q : \nabla v \; + \; q \cdot v) \; dx \; - \; \sum_{j=1}^{N_{\rm f}} \int_{\mathtt{F}_j} \big\{\nabla q \mathbf{n}^+\big\} \cdot J(v) + \big\{\nabla v \mathbf{n}^+\big\} \cdot J(q) - \eta \{\mathtt{r}_j(J(q))\} \cdot J(v) \, dx \end{cases} \tag{36a}$$

where $\mathtt{r}_j : [L^2(\mathtt{F}_j)]^D \to \mathcal{X}_{\rm hf}$ is the BR2 lifting operator (cf. [6]) given by

$$(\mathtt{r}_j(w), v) = -\int_{\mathtt{F}_j} w \cdot \{v\} \, dx \quad \forall\, w \in [L^2(\mathtt{F}_j)]^D, \;\; v \in \mathcal{X}_{\rm hf} \quad j = 1, \ldots, N_{\rm f}, \tag{36b}$$

and $\eta > 0$ is a stabilization parameter that is here set equal to $d+1$. In the remainder of this section, we discuss the construction of the various pieces of the formulation.

5.1 Online solution method

We denote by $\{\psi_i\}_{i=1}^m$ an orthonormal basis of $\widehat{\mathcal{Y}}$; we introduce the set of indices $\mathtt{I}_{\rm eq,e} = \{k \in \{1,\ldots,N_{\rm e}\} : \rho^{\rm eq,e}_k \neq 0\}$ and $\mathtt{I}_{\rm eq,f} = \{j \in \{1,\ldots,N_{\rm f}\} : \rho^{\rm eq,f}_j \neq 0\}$. Then, we rewrite the minimization statement in (35a) as the nonlinear least-square problem

$$\min_{\boldsymbol{\alpha} \in \mathbb{R}^n} \left\| \mathfrak{R}^{\rm eq}_\mu(\boldsymbol{\alpha}) \right\|_2, \quad \text{with} \;\; \big(\mathfrak{R}^{\rm eq}_\mu(\boldsymbol{\alpha})\big)_i = \mathfrak{R}^{\rm eq}_\mu(\mathtt{Z}\boldsymbol{\alpha}, \psi_i), \;\; i = 1, \ldots, m, \tag{37}$$

which can be solved using the Gauss-Newton method (GNM). Note that the computation of the entries of $\mathfrak{R}^{\rm eq}_\mu(\boldsymbol{\alpha})$ for any $\boldsymbol{\alpha} \in \mathbb{R}^n$ requires to compute the local elemental residuals $\{r^{\rm e}_{k,\mu}\}_k$ for all $k \in \mathtt{I}_{\rm eq,e}$ and the facet residuals $\{r^{\rm f}_{j,\mu}\}_j$ for all $j \in \mathtt{I}_{\rm eq,f}$; towards this end, we should store the trial and test ROBs in the sampled elements

$$\Omega_{\rm eq} := \left(\bigcup_{k \in \mathtt{I}_{\rm eq,e}} \mathtt{D}_k \right) \cup \left(\bigcup_{j \in \mathtt{I}_{\rm eq,f}} \mathtt{D}_j^+ \cup \mathtt{D}_j^- \right). \tag{38}$$

We conclude that online storage and computational costs scale linearly with the cardinality of $|\mathtt{I}_{\rm eq,e}|$ and $|\mathtt{I}_{\rm eq,f}|$.

Our formulation enables a straightforward *discretize-then-map* treatment of geometry variations: the elemental residual $r^{\rm e}_{k,\mu}(\cdot,\cdot)$ depends on the nodes $\{x^{\rm hf}_{\mathtt{T}_{i,k}}\}_{i=1}^{n_{\rm lp}}$ of the k-th element of the mesh; given a new value of the parameter μ, it hence suffices to deform the nodes of the sampled elements through the mapping Φ_μ *before* starting the GNM iterations. Similar reasoning applies to the facet integrals. As discussed in [54] this approach enables the use of the routines of the DG HF code and is thus simple to implement.

Several variants of the present approach are available in the literature. In [54], Yano considered an elementwise EQ procedure that guarantees relevant conservation properties, while in [17] Du and Yano proposed a pointwise EQ procedure that generates sparse quadrature rules within each element and facet. Our approach enables slightly larger reductions than the approach in [54] and, unlike the approach in [17] can cope with elementwise terms such as the BR2 lifting operator (see (36b)) or elementwise artificial viscosities of the form (10). A thorough comparison of our method with other EQ formulations is beyond the scope of this work.

We finally comment on the choice of the initial condition for GNM. We here rely on nearest-neighbor regression: given the training set of simulations $\{\widetilde{q}^{\rm hf}_\mu : \mu \in \mathcal{P}_{\rm train}\}$, we define the corresponding best-fit generalized coordinates $\{\boldsymbol{\alpha}^{\rm bf}_\mu : \mu \in \mathcal{P}_{\rm train}\}$ obtained by projecting the available snapshots on the ROB $\mathtt{Z}$; then, for any $\mu \in \mathcal{P}$, we initialize GNM with $\boldsymbol{\alpha}^{\rm bf}_{\mu_{\rm nn}}$ with $\mu_{\rm nn} = \arg\min_{\mu' \in \mathcal{P}_{\rm train}} \|\mu - \mu'\|_2$. We observe that the present approach might be highly suboptimal if the cardinality of $\mathcal{P}_{\rm train}$ is modest: in section 6, we discuss how to improve the initialization of GNM using information from the previous iterations of Algorithm 1.

5.2 Construction of the empirical test space

As in [20], we here resort to the sampling strategy based on proper orthogonal decomposition (POD, [41, 51]) proposed in [48] to construct the test space $\widehat{\mathcal{Y}}$ in (35). Given the training set $\mathcal{P}_{\text{train}} = \{\mu^k\}_{k=1}^{n_{\text{train}}} \subset \mathcal{P}$, the associated snapshots $\{\widetilde{q}_\mu^{\text{hf}} : \mu \in \mathcal{P}_{\text{train}}\}$, and the trial ROB $\{\zeta_i\}_{i=1}^n$, we compute the test snapshot set

$$((\psi_{k,i}, v)) \,=\, \mathfrak{J}_{\mu^k}^{\text{hf}}[\widetilde{q}_{\mu^k}^{\text{hf}}](\zeta_i, v), \quad \forall\, v \in \mathcal{X}_{\text{hf}},$$

for $i = 1, \ldots, n$ and $k = 1, \ldots, n_{\text{train}}$, where $\mathfrak{J}_\mu^{\text{hf}}[q] : \mathcal{X}_{\text{hf}} \times \mathcal{X}_{\text{hf}} \to \mathbb{R}$ denotes the Fréchet derivative of the HF residual at q. Then, we perform POD on the test snapshot set $\{\psi_{k,i}\}_{k,i}$ based on the $((\cdot,\cdot))$ inner product (36) to obtain $\widehat{\mathcal{Y}}$. In all the numerical experiments, we consider test spaces of size $j_{\text{es}} = \dim(\widehat{\mathcal{Y}}) = 2n$; alternatively, we might choose the dimension of $\widehat{\mathcal{Y}}$ using an energy criterion. We refer to [48, Appendix C] for a rigorous justification of our method for linear inf-sup stable problems.

5.3 Hyper-reduction

We seek $\boldsymbol{\rho}^{\text{eq,e}} \in \mathbb{R}_+^{N_{\text{e}}}$ and $\boldsymbol{\rho}^{\text{eq,f}} \in \mathbb{R}_+^{N_{\text{f}}}$ in (35b) such that

(i) (*efficiency constraint*) the number of nonzero entries in $\boldsymbol{\rho}^{\text{eq,e}}, \boldsymbol{\rho}^{\text{eq,f}}$, $\|\boldsymbol{\rho}^{\text{eq,e}}\|_{\ell^0}$ and $\|\boldsymbol{\rho}^{\text{eq,f}}\|_{\ell^0}$, is as small as possible;

(ii) (*constant function constraint*) the constant function is approximated correctly in Ω (for $\Phi = \texttt{id}$),

$$\Big|\sum_{k=1}^{N_{\text{e}}} \rho_k^{\text{eq,e}} |\mathtt{D}_k| \,-\, |\Omega|\Big| \ll 1, \quad \Big|\sum_{j=1}^{N_{\text{f}}} \rho_j^{\text{eq,f}} |\mathtt{F}_j| \,-\, \sum_{j=1}^{N_{\text{f}}} |\mathtt{F}_j|\Big| \ll 1; \tag{39}$$

(iii) (*manifold accuracy constraint*) for all $\mu \in \mathcal{P}_{\text{train,eq}} = \{\mu^k\}_{k=1}^{n_{\text{train}}+n_{\text{train,eq}}}$, the empirical residual satisfies

$$\Big\|\mathfrak{R}_\mu^{\text{hf}}(\boldsymbol{\alpha}_\mu^{\text{train}}) \,-\, \mathfrak{R}_\mu^{\text{eq}}(\boldsymbol{\alpha}_\mu^{\text{train}})\Big\|_2 \ll 1. \tag{40a}$$

where $\mathfrak{R}_\mu^{\text{hf}}$ corresponds to substitute $\rho_1^{\text{eq,e}} = \ldots = \rho_{N_{\text{e}}}^{\text{eq,e}} = \rho_1^{\text{eq,f}} = \ldots = \rho_{N_{\text{f}}}^{\text{eq,f}} = 1$ in (35b) and $\boldsymbol{\alpha}_\mu^{\text{train}}$ satisfies

$$\boldsymbol{\alpha}_\mu^{\text{train}} = \begin{cases} \arg\min\limits_{\boldsymbol{\alpha}\in\mathbb{R}^n} \|\mathtt{Z}\boldsymbol{\alpha} - \widetilde{q}_\mu^{\text{hf}}\|_2, & \text{if } \mu \in \mathcal{P}_{\text{train}}; \\ \arg\min\limits_{\boldsymbol{\alpha}\in\mathbb{R}^n} \|\mathfrak{R}_\mu^{\text{hf}}(\boldsymbol{\alpha})\|_2, & \text{if } \mu \notin \mathcal{P}_{\text{train}}; \end{cases} \tag{40b}$$

and $\mathcal{P}_{\text{train}} = \{\mu^k\}_{k=1}^{n_{\text{train}}}$ is the set of parameters for which the HF solution is available.

We refer to the above-mentioned literature for a thorough motivation of the previous constraints. We remark that several authors (see [54, Algorithm 1]) have observed that considering an augmented training set $\mathcal{P}_{\text{train,eq}}$ in (40) might improve performance of the hyper-reduced ROM, particularly for small values of n_{train}. However, for the numerical experiments of this work, we empirically observed that the choice $\mathcal{P}_{\text{train}} = \mathcal{P}_{\text{train,eq}}$ leads to accurate results.

It is possible to show (see, e.g., [48]) that (i)-(ii)-(iii) lead to a sparse representation problem of the form

$$\min_{\boldsymbol{\rho}\in\mathbb{R}^{N_{\text{e}}+N_{\text{f}}}} \|\boldsymbol{\rho}\|_{\ell^0}, \quad \text{s.t} \quad \begin{cases} \|\mathbf{G}\boldsymbol{\rho} - \mathbf{b}\|_2 \le \delta; \\ \boldsymbol{\rho} \ge \mathbf{0}; \end{cases} \tag{41}$$

for a suitable threshold $\delta > 0$, and for a suitable choice of $\mathbf{G}, \mathbf{b}$. Following [18], we here resort to the non-negative least-squares method to find approximate solutions to (41). In particular, we use the Matlab function `lssnonneq`, which takes as input the pair $(\mathbf{G}, \mathbf{b})$ and a tolerance $tol_{\text{eq}} > 0$ and returns the sparse vectors $\boldsymbol{\rho}^{\text{eq,e}}, \boldsymbol{\rho}^{\text{eq,f}}$,

$$[\boldsymbol{\rho}^{\text{eq,e}}, \boldsymbol{\rho}^{\text{eq,f}}] = \texttt{lsqnonneg}\,(\mathbf{G}, \mathbf{b}, tol_{\text{eq}})\,. \tag{42}$$

We refer to [12] for an efficient implementation of the non-negative least-squares method for large-scale problems.

5.4 Construction of the trial space via greedy sampling

We resort to the weak-greedy algorithm (cf. [50]) to build the ROM and the trial ROB $\mathtt{Z}$; the weak-greedy method relies on the repeated maximization of an error indicator to adaptively sample the parameter domain;

Algorithm 2 summarizes the overall procedure, while Algorithm 3 summarizes the construction of the ROM. In this work, we consider the residual-based error indicator (cf. [20, section 3.2.3]),

$$\Delta : \mu \in \mathcal{P} \mapsto \sup_{v \in \mathcal{X}_{\mathrm{hf}}} \frac{\mathfrak{R}^{\mathrm{hf}}_{\mu}(\widetilde{q}_{\mu}, v)}{|||v|||}. \tag{43}$$

Note that the evaluation of (43) requires the solution to a linear system of size N_{hf}: it is hence ill-suited for real-time online computations; nevertheless, in our experience the offline cost associated with the evaluation of (43) is a fraction of the cost to perform hyper-reduction and to build the test space $\widehat{\mathcal{Y}}$. We refer to [20] and to the references therein for a thorough discussion on the construction of an inexpensive surrogate of (43). Even if we empirically observe that our residual-based error indicator is highly-correlated with the true error, it does not provide a rigorous bound; for this reason, after having computed the new HF solution (cf. Line 5, Algorithm 2) we check if the relative error is below a given threshold for the parameter that maximizes the error indicator.

Algorithm 2 : weak-greedy algorithm.

Inputs: $\mathcal{P}_{\mathrm{train,gr}} := \{\mu^k_{\mathrm{gr}}\}_{k=1}^{n_{\mathrm{train}}}$ training parameter set, $\Phi : \Omega \times \mathcal{P} \to \Omega$ mapping; $\mathcal{T}_{\mathrm{hf}}$ mesh.
Outputs: `Z` trial ROB; $\mu \in \mathcal{P} \mapsto \widehat{\boldsymbol{\alpha}}_{\mu}$ ROM for the solution coefficients.

1: Choose $\mathcal{P}_\star = \{\mu^{\star,i}\}_{i=1}^{n_0}$ and compute the HF solutions $\mathcal{S}_\star = \{\widetilde{q}^{\mathrm{hf}}_{\mu} : \mu \in \mathcal{P}_\star\}$.
2: **for** $n = n_0 + 1, \ldots, n_{\max}$ **do**
3: Update the ROB `Z` and the ROM (cf. Algorithm 3).
4: Estimate the solution $\widetilde{q}^{\mathrm{hf}}_{\mu}$ and compute the indicator Δ_μ in (43) for all $\mu \in \mathcal{P}_{\mathrm{train,gr}}$.
5: Compute $\widetilde{q}^{\mathrm{hf}}_{\mu^{\star,n}}$ for $\mu^{\star,n} = \arg\max_{\mu \in \mathcal{P}_{\mathrm{train,gr}}} \Delta_\mu$; update $\mathcal{P}_\star$ and $\mathcal{S}_\star$.
6: **if** $\|\widetilde{q}^{\mathrm{hf}}_{\mu^{\star,n}} - \widetilde{q}_{\mu^{\star,n}}\| < \mathtt{tol} \|\widetilde{q}^{\mathrm{hf}}_{\mu^{\star,n}}\|$ **then**
7: Update the ROB `Z` and the ROM.
8: `break`
9: **end if**
10: **end for**

Algorithm 3 : construction of the ROM.

Inputs: snapshot set $\mathcal{S}_\star := \{(\mu, \widetilde{q}^{\mathrm{hf}}_{\mu}) : \mu \in \mathcal{P}_\star\}$.
Outputs: `Z` trial ROB; $\mu \in \mathcal{P} \mapsto \widehat{\boldsymbol{\alpha}}_{\mu}$ ROM for the solution coefficients.

1: Define the test space $\widehat{\mathcal{Y}}$ (cf. section 5.2)
2: Define the EQ weights $\boldsymbol{\rho}^{\mathrm{eq,e}}, \boldsymbol{\rho}^{\mathrm{eq,f}}$ (cf. section 5.3).
3: Store trial and test ROBs, and grid points in the reduced mesh (cf. (38)).

We observe that the weak-greedy algorithm requires multiple definitions of the ROM, which imply multiple constructions of the test space $\widehat{\mathcal{Y}}$, the quadrature weights $\boldsymbol{\rho}^{\mathrm{eq,e}}, \boldsymbol{\rho}^{\mathrm{eq,f}}$ and multiple greedy searches over the training set $\mathcal{P}_{\mathrm{train,gr}}$ (cf. Line 4, Algorithm 2). As reported in Table 1, the overhead costs of the greedy procedure — that is, the total cost of the procedure minus the cost of the HF solves — might be significant. This observation motivates the development of more sophisticated training strategies to reduce offline costs. We address this issue in section 6.

6 Adaptive procedure

Each iteration of Algorithm 1 generates a large amount of data about the parametric problem, which can be used to speed up offline computations. In the remainder of this section, we illustrate computational bottlenecks of the training phase and we discuss actionable strategies to reduce the computational burden; in the numerical investigations, we assess the impact of these choices.

- The construction of the snapshot set for registration (cf. Line 3, Algorithm 1) based on the HF model is prohibitively expensive. Instead, we propose to rely on the ROM built at the previous iteration; for the first iteration, we first execute the weak-greedy algorithm and then we use the ROM to generate the dataset of simulations.

- The GNM for (37) is sensitive to the choice of the initial condition. In our implementation, we initialize GNM based on nearest-neighbor regression which is clearly highly inaccurate for modest values of n. To face this issue, we propose to rely on a large dataset of initial conditions defined as follows:

$$\left\{\widehat{\boldsymbol{\alpha}}^0_\mu : \mu \in \mathcal{P}_{\text{train}}\right\} \quad \text{where } \widehat{\boldsymbol{\alpha}}^0_\mu = \arg\min_{\boldsymbol{\alpha}\in\mathbb{R}^n} \|\mathtt{Z}\boldsymbol{\alpha} - \widehat{q}^{\text{old}}_\mu \circ \Phi_\mu^{-1}\|. \tag{44}$$

  Note that the fields $\{\widehat{q}^{\text{old}}_\mu : \mu \in \mathcal{P}_{\text{train}}\}$ are generated for registration (cf. Line 3, Algorithm 1); nevertheless, cost of (44) is significant due to the need to compute the composition of $\widehat{q}^{\text{old}}_\mu$ with Φ_μ^{-1} — which requires mesh interpolation. In practice, we estimate the L^2 norm $\|\cdot\|$ in (44) using 10^3 randomly-sampled points in Ω to reduce offline costs.

- The PTC strategy employed to solve the HF problem (cf. section 2) might require many iterations to reach convergence. To reduce the computational burden, we initialize the PTC solver with the reduced-order solution $\widehat{q}_\mu$ from the previous iteration, as opposed to the free-stream solution. Thanks to this choice, we can consider a much larger initial CFL number[3] without experiencing any stability issue.

- The registration procedure discussed in section 4 relies on multiple solutions to a nonlinear non-convex optimization problem of size $m = \mathcal{O}(10^2)$ that is sensitive to the initial condition. In our experience, the initialization strategy reviewed in section 4.3 leads to accurate performance; however, it requires a sufficiently dense discretization of $\mathcal{P}$ and is not parallelizable. To address this issue, we propose to store the mapping coefficients $\{\widehat{\mathbf{a}}_\mu : \mu \in \mathcal{P}_{\text{train}}\}$ obtained during the first iteration of the registration method and then use them as initial conditions for the subsequent iteration: note that for this choice of the initialization the solution to the problems (34) can be trivially parallelized; in addition, we can potentially cope with much coarser discretizations of $\mathcal{P}$.

- As discussed in section 5, the weak-greedy algorithm requires multiple constructions of the ROM and might hence be expensive; in addition, it cannot be efficiently parallelized. To address this issue, we initialize Algorithm 2 with the parameters $\{\mu^{\star,i}\}_{i=1}^{n_0}$ obtained by applying the strong-greedy algorithm to the snapshot set generated for registration (cf. Line 3, Algorithm 1). Since the snapshot set is generated using the ROM, the strong-greedy algorithm can be applied to the generalized coordinates. For completeness, we report the strong-greedy procedure in Appendix A.

7 Numerical results

We present below extensive numerical investigations for the model problems introduced in section 2.2. Further numerical tests are provided in Appendix B. We assess performance based on $n_{\text{test}} = 20$ out-of-sample parameters $\mathcal{P}_{\text{test}} = \{\mu^j_{\text{test}}\}_{j=1}^{n_{\text{test}}}$ with $\mu^1_{\text{test}}, \dots, \mu^{n_{\text{test}}}_{\text{test}} \overset{\text{iid}}{\sim} \text{Uniform}(\mathcal{P})$; for each $\mu \in \mathcal{P}_{\text{test}}$, we report the HF L^2 error E^{hf}_μ, the sub-optimality index η^{hf}_μ and the total enthalpy error E^∞_μ such that

$$E^{\text{hf}}_\mu = \frac{\|q^{\text{hf}}_\mu - \widehat{q}^{\text{hf}}_\mu\|_{L^2(\Omega_\mu)}}{\|q^{\text{hf}}_\mu\|_{L^2(\Omega_\mu)}}, \tag{45}$$

$$\eta^{\text{hf}}_\mu = \frac{\|q^{\text{hf}}_\mu - \widehat{q}^{\text{hf}}_\mu\|_{L^2(\Omega_\mu)}}{\min_{\zeta\in\mathcal{Z}_n} \|q^{\text{hf}}_\mu - \zeta \circ \Phi_\mu^{-1}\|_{L^2(\Omega_\mu)}}, \tag{46}$$

$$E^\infty_\mu = \frac{\|H^{\text{true}}_{\text{tot},\mu} - \widehat{H}_{\text{tot},\mu}\|_{L^2(\Omega_\mu)}}{\|H^\infty_{\text{tot},\mu}\|_{L^2(\Omega_\mu)}}. \tag{47}$$

The relative error E^{hf}_μ measures the accuracy of the reduced-order estimate with respect to the HF model employed for training — it is hence a measure of the overall ability of the MOR procedure to approximate the truth model of the PDE. The suboptimality index η^{hf}_μ measures the extent to which the LSPG projection scheme is suboptimal compared to the best-fit error: it hence allows to directly evaluate the effectiveness of the ROM, which encompasses the choice of the test space, initialization, and hyper-reduction. Finally, the total enthalpy error (47) measures the accuracy of the state estimate with respect to the exact solution to the PDE, in terms of enthalpy preservation. Simulations are performed in Matlab 2022a [31] based on an in-house code, and executed over a commodity Linux workstation (RAM 32 GB, Intel i7 CPU 3.20 GHz x 12).

[3] In the numerical experiments, we set $\text{CFL}_0 = 100$ instead of $\text{CFL}_0 = 1$; see [56, section II.B].

7.1 Inviscid flow through a nozzle

We perform $N_{\text{it}} = 3$ iterations of Algorithm 1 without acceleration. We initialize the algorithm using an uniform HF grid with $N_{\text{e}} = 60$ triangles and quadratic ($\texttt{p} = 2$) polynomials; then, we increase the size of the mesh by a factor 1.5 at each iteration: this implies that the generated HF meshes have $N_{\text{e}} = 60$, $N_{\text{e}} = 90$ and $N_{\text{e}} = 135$ elements at iterations one, two and three, respectively. We consider a regular 15×15 grid of parameters $\mathcal{P}_{\text{train}}$ for registration and a regular 10×10 grid of parameters $\mathcal{P}_{\text{train,gr}}$ in Algorithm 2. We rely on the HF solver to generate the dataset of simulations at iteration one, while we rely on the ROM from previous iterations to generate the snapshot set (cf. Line 3, Algorithm 1) for $k = 2, \dots, N_{\text{it}}$. We consider the tolerance $\texttt{tol} = 10^{-3}$ in Algorithm 2 and we consider an initial regular 3×3 grid of parameters to initialize the ROM: the algorithm generates ROBs of size $n = 15$, $n = 20$ and $n = 10$.

Figure 3 shows the performance of the ROM. Figure 3(a) shows the relative error over the test set, which mildly depends on the size of the mesh. Figure 3(b) shows the suboptimality index: interestingly, we observe that the performance of the projection scheme deteriorates as we increase the size of the mesh: we plan to investigate this behavior in a subsequent work; nevertheless, we observe that $\eta_\mu^{\text{hf}} \lesssim 10$ for all numerical experiments. Figure 3(c) shows the total enthalpy error: as expected, the error decreases as we increase the size of the mesh. Figure 3(d) shows the wall-clock online cost: thanks to hyper-reduction, results do not depend on the size of the underlying mesh but they clearly depend on the size n of the ROB.

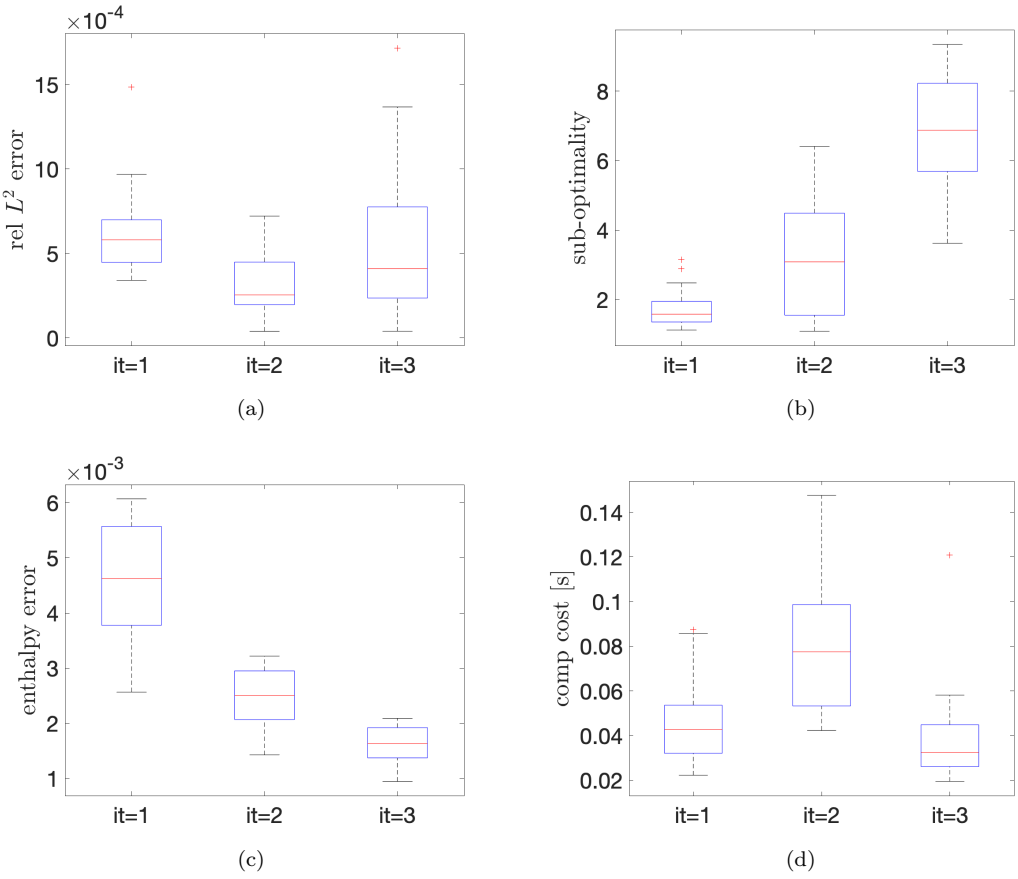
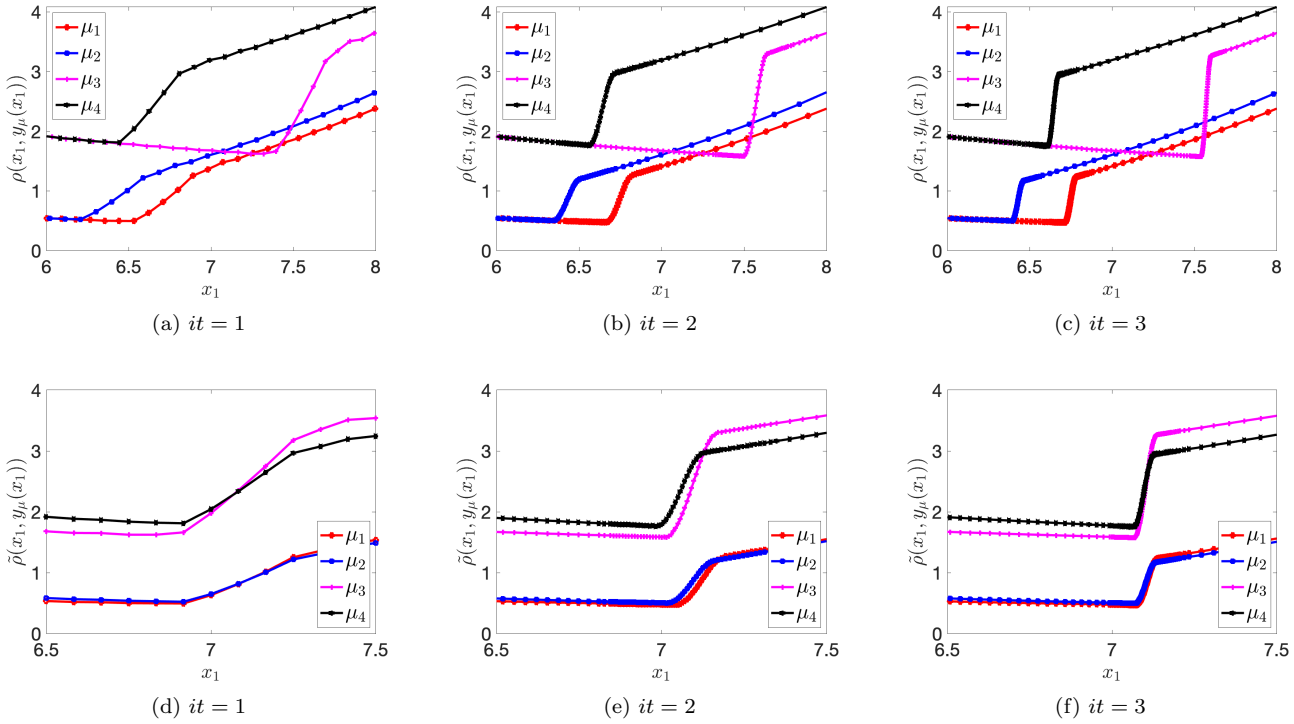


Figure 3: nozzle flow. Performance of the ROM for three iterations of the adaptive (basic) procedure.

Figure 4 shows the behavior of the modified density $\bar{\rho} := A\rho$ in the proximity of the shock for four parameter values, for three iterations of the algorithm, and for both physical and reference configurations. We observe that registration is effective to track the position of the discontinuity.

In Figure 5, we investigate the effect of registration on solution manifold compressibility and mesh adaptation. Towards this end, we consider the adaptive reduced-order and HF models associated with the third iteration of Algorithm 1, and a HF model defined over a Cartesian "static" mesh with the same number of elements, $N_{\text{e}} = 135$. First, in Figure 5(a), we compare the behavior of the normalized POD eigenvalues associated with the snapshot set in physical ("unreg") and reference ("reg") configurations. We observe that registration significantly improves the convergence of the POD eigenvalues that can be regarded as a "proxy" of the linear complexity of the corresponding solution manifold. Figure 5(b) shows the behavior of the error in total enthalpy for the final registered ROM and the static HF model based on an uniform mesh: we clearly notice that the HF

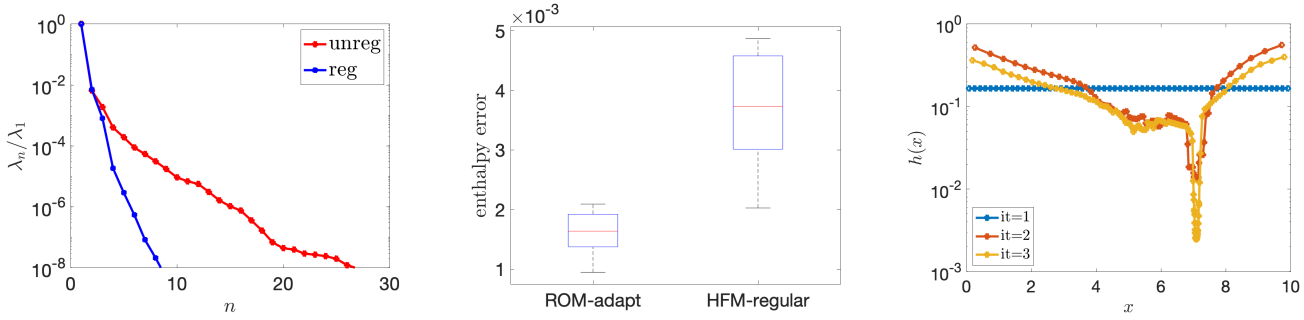
model — which has the same number of degrees of freedom as the HF model used to generate the ROM — is significantly less accurate than the adapted ROM. Finally, Figure 5(c) shows the behavior of the mesh density $h : \Omega \to \mathbb{R}_+$ such that $h|_{\mathtt{D}_k} = |\mathtt{D}_k|$ for the sequence of meshes generated during Algorithm 1: we observe that registration allows us to refine the mesh over a very narrow portion of the computational domain and hence enables significant computational savings.



(a) $it = 1$ (b) $it = 2$ (c) $it = 3$

(d) $it = 1$ (e) $it = 2$ (f) $it = 3$

Figure 4: nozzle flow. Behavior of the (modified) density field in physical (cf. (a)-(b)-(c)) and reference (cf. (d)-(e)-(f)) configuration for four values of the parameter and three iterations of the adaptive algorithm (basic version).



(a) (b) (c)

Figure 5: nozzle flow. Effect of registration on compressibility and mesh adaptation. (a) POD eigenvalues in reference and physical configurations. (b) total enthalpy error for registered ROM and unregistered HFM on regular mesh with $N_\mathrm{e} = 135$ elements. (c) mesh density $h : \Omega \to \mathbb{R}_+$, $h|_{\mathtt{D}_k} = |\mathtt{D}_k|$ for the sequence of considered meshes.

7.2 Inviscid flow over a Gaussian bump

We perform $N_\mathrm{it} = 3$ iterations of Algorithm 1 without and with acceleration; we consider both isotropic and anisotropic mesh adaptation based on the software `mmg2d` and on the metrics introduced in section 3. As in the previous case, we rely on a regular 15×15 grid of parameters $\mathcal{P}_\mathrm{train}$ for registration and a regular 10×10 grid of parameters $\mathcal{P}_\mathrm{train,gr}$ in Algorithm 2. We set $\mathtt{tol} = 10^{-3}$ in the termination condition of Algorithm 2. To reduce training costs of the first snapshot generation, we first perform a weak-greedy algorithm to generate

a ROM that is later used to generate the snapshot set. In all our tests, we consider the initial grid depicted in Figure 8(a) with $N_e = 3448$, and we rely on a quadratic approximation. We state upfront that the registration algorithm returns a low-rank mapping with $m = 2$ modes for all runs considered.

7.2.1 Basic approach

We first study the performance of the standard (without acceleration) approach based on isotropic mesh adaptation. Figure 6 replicates the results in Figure 3 for the transonic bump problem. We observe that the ROM achieves accurate performance over the test set with respect to the HF estimate for all three iterations: results are hence in good agreement with the selected tolerance ($\texttt{tol} = 10^{-3}$) of Algorithm 2. The suboptimality index ranges from one to three for all experiments: this indicates that our projection scheme is extremely effective for this model problem. The total enthalpy error decreases as we increase the size of the mesh, while the computational cost is nearly the same for all iterations.

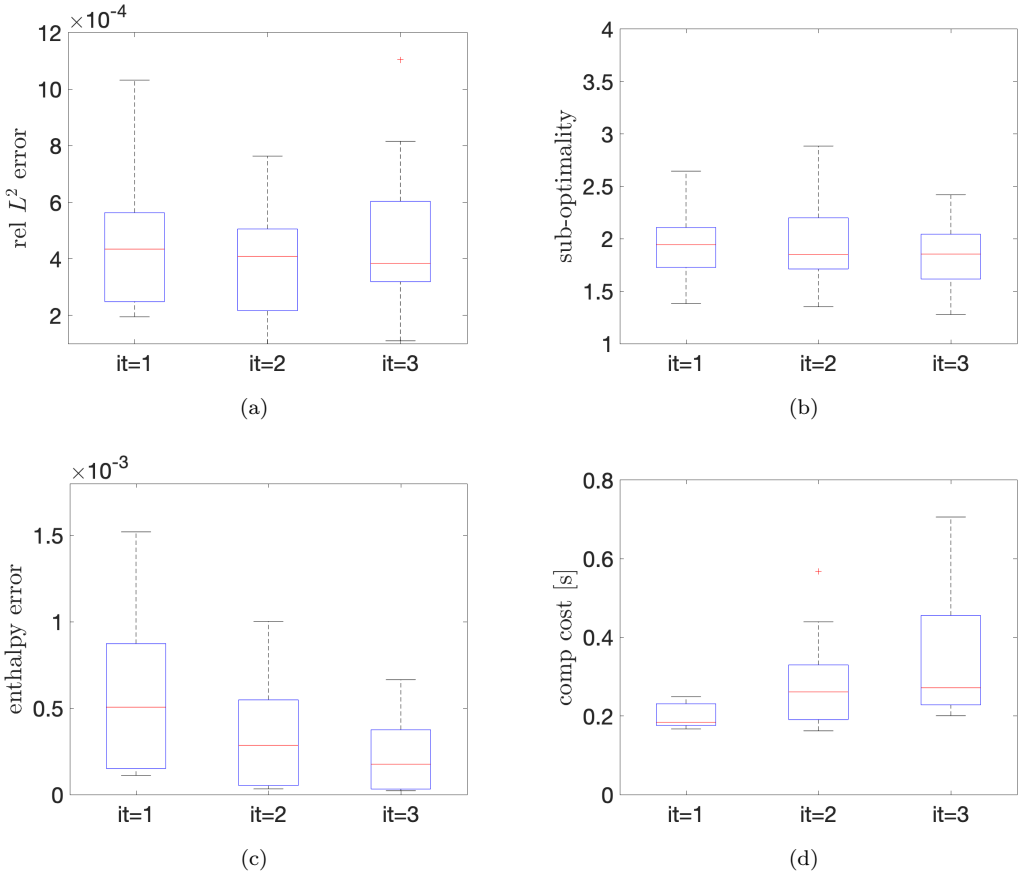
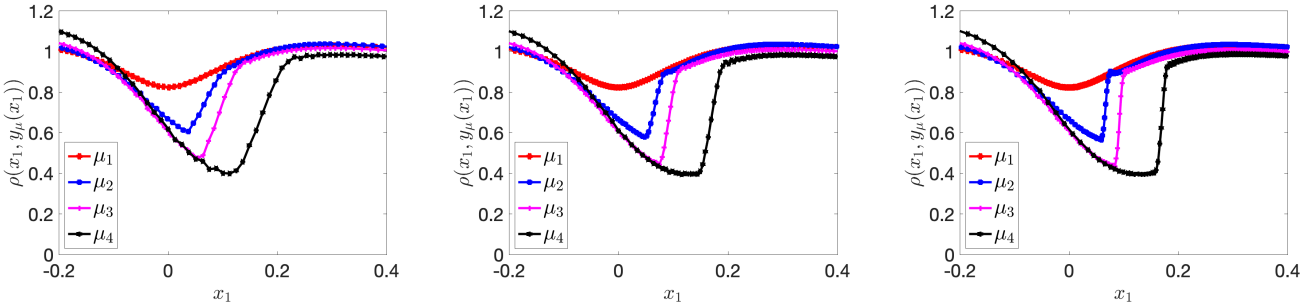
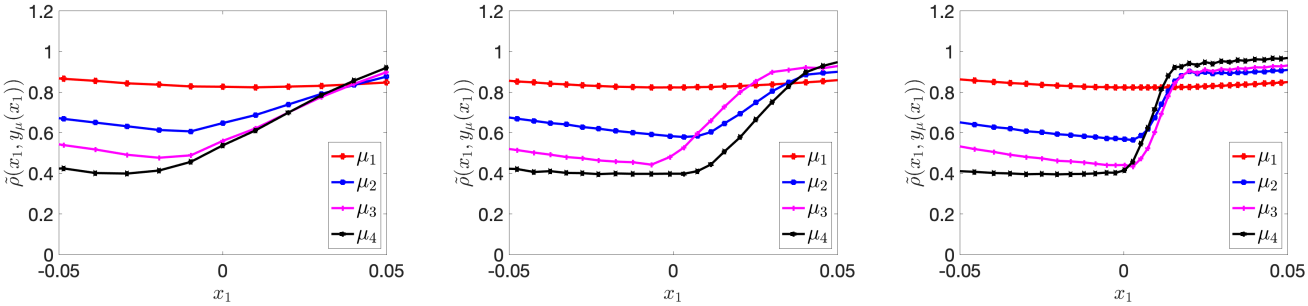


Figure 6: transonic bump. Performance of the ROM for three iterations of the adaptive (basic) procedure.

Figure 7 shows the behavior of the density field over the bump in physical and reference configuration for four values of the parameter and three iterations of the adaptive algorithm. We clearly notice the effect of the registration to nearly "freeze" the position of the shock — when present — in the reference configuration. We also notice that mesh adaptation is effective to sharpen the approximation of the shock as we increase the size of the mesh.



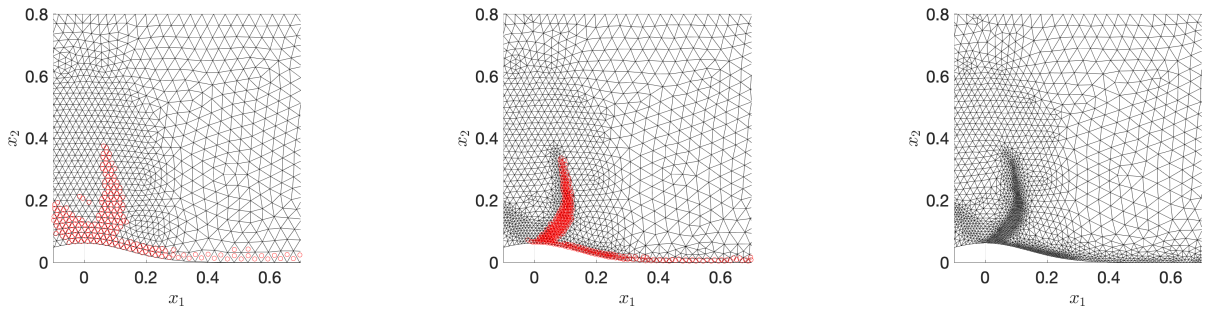
(a) $it = 1$ (b) $it = 2$ (c) $it = 3$



(d) $it = 1$ (e) $it = 2$ (f) $it = 3$

Figure 7: transonic bump. Behavior of the (modified) density field in physical (cf. (a)-(b)-(c)) and reference (cf. (d)-(e)-(f)) configuration for four values of the parameter and three iterations of the adaptive algorithm (basic version).

Figure 8 shows the reference mesh in the proximity of the bump, for three iterations of the adaptive algorithm; red dots indicate the centers of the marked elements at iterations $it = 2$ and $it = 3$. Interestingly, we observe that the mesh is adapted in the proximity of the shock and in the proximity of the lower wall, in the area downstream of the bump: as for the previous example, registration facilitates the task of parametric mesh adaptation by "freezing" the coherent flow structure in the reference domain.



(a) $it = 1$ (b) $it = 2$ (c) $it = 3$

Figure 8: transonic bump. Visualization of the reference mesh in the proximity of the bump, for three iterations of the adaptive algorithm (basic). Red dots indicate the centers of the marked elements.

Table 1 provides an overview of the offline costs. We notice that the costs are dominated by snapshot generation at iteration $it = 1$ — which involves the construction of the ROM — and by the weak-greedy algorithms. We also observe that in our implementation the overhead costs of the greedy method are significant, while the cost of mesh adaptation is completely negligible. For the registration algorithm, we distinguish between the cost to estimate the sensors $\{s_\mu^{\text{hf}} : \mu \in \mathcal{P}_{\text{train}}\}$ in (33) (cf. Remark 4.1) and the cost of solving the parametric registration problem: the former involves mesh interpolation over a curved HF mesh and is embarrassingly parallel, while the latter is dominated by the solution to the optimization problems (34) for the first iteration.

	$it=1$		$it=2$		$it=3$	
	basic	acc.	basic	acc.	basic	acc.
avg nbr its	18.64	8.28	20.53	8.71	29.00	8.62
avg cost	54.44	25.57	90.58	42.66	208.48	68.26

Table 2: performance of the HF solver for two initialization strategies.

	it = 1	it = 2	it = 3
ROB size:	14	17	19
mesh size:	3448	4653	6667
snapshot generation:	1459.37	43.86	59.59
registration (sensor def.):	197.44	202.13	216.09
registration (optimization):	402.93	508.71	770.13
mesh adaptation:	0.00	0.35	0.57
greedy alg (HF solves):	762.15	1539.81	3961.10
greedy alg (overhead):	295.36	632.19	1200.36

Table 1: transonic bump. Offline training costs (in seconds) of the adaptive (basic) approach.

7.2.2 Acceleration of training through multi-fidelity strategies

We investigate the effect of the acceleration strategy discussed in section 6; to facilitate the comparison with the results of the previous section, we consider isotropic mesh adaptation. First, Table 2 investigates the effect of the initialization strategy on the convergence of the HF solver; the computational cost includes the interpolation cost. We observe that our initialization strategy reduces the number of iterations required for convergence by roughly a factor three and computational costs by roughly a factor three for the final iteration.

Table 3 provides an overview of the two approaches in terms of the two metrics (45) and (47). Note that the acceleration strategy reduces offline costs by roughly 30% mostly due to the reduction of the cost of the HF solves; it also slightly reduces the online costs by providing a more accurate initialization for GNM. Further numerical investigations are provided in Appendix B. We insist that the current implementation does not exploit parallel computing: since the acceleration strategy enables a much more efficient parallelization (cf. section 6), we expect more significant gains for the accelerated procedure when combined with parallel computing.

	L^2 error (avg)			enthalpy error (avg)		
	1	2	3	1	2	3
Basic	$0.53 \cdot 10^{-3}$	$0.44 \cdot 10^{-3}$	$0.48 \cdot 10^{-3}$	$0.58 \cdot 10^{-3}$	$0.35 \cdot 10^{-3}$	$0.23 \cdot 10^{-3}$
Accelerated	$0.29 \cdot 10^{-3}$	$0.31 \cdot 10^{-3}$	$0.49 \cdot 10^{-3}$	$0.59 \cdot 10^{-3}$	$0.37 \cdot 10^{-3}$	$0.23 \cdot 10^{-3}$

	ROB size			online cost (avg)			offline cost
	1	2	3	1	2	3	
Basic	14	17	19	0.20	0.28	0.36	03:24:13
Accelerated	18	17	21	0.25	0.25	0.36	02:11:47

Table 3: Comparison of the performance of the basic and accelerated adaptive procedures.

Figure 9 investigates the performance of the greedy strategy. We perform the strong-greedy algorithm on the snapshot sets generated at iterations one, two and three to identify the "optimal" parameters $\mathcal{P}_\star^{it,n} = \{\mu^{\star,it,i}\}_{i=1}^n$ for $it = 1, 2, 3$. Then, we compute the projection error

$$E_{\text{proj},\mu} := \frac{\min_{\zeta \in \mathcal{Z}_\star^{it,n}} \|\widetilde{q}_\mu^{\text{hf}} - \zeta\|}{\|\widetilde{q}_\mu^{\text{hf}}\|}, \quad \text{where } \mathcal{Z}_\star^{it,n} = \text{span}\left\{\widetilde{q}_\mu^{\text{hf}} \ : \ \mu \in \mathcal{P}_\star^{it,n}\right\}$$

over the test set of $n_{\text{test}} = 20$ simulations; here, $\widetilde{q}_\mu^{\text{hf}}$ refers to the HF estimate obtained using the DG model at the final ($it = 3$) iteration. To provide a concrete reference, we compare the results obtained using regular grids ($2 \times 2, 3 \times 3, 4 \times 4$) of parameters. We observe that the strong greedy algorithm based on iteration $it = 2$ provides results that are nearly as good as the results obtained based on the snapshot set of iteration $it = 3$. This empirical finding suggests that the application of the strong-greedy method to a lower-fidelity snapshot set might provide an inexpensive yet effective sampling strategy for model reduction.

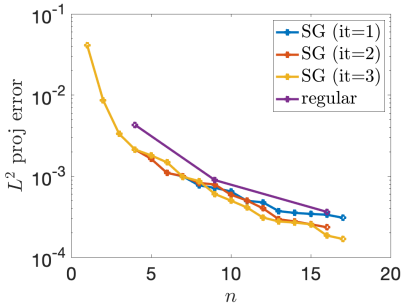
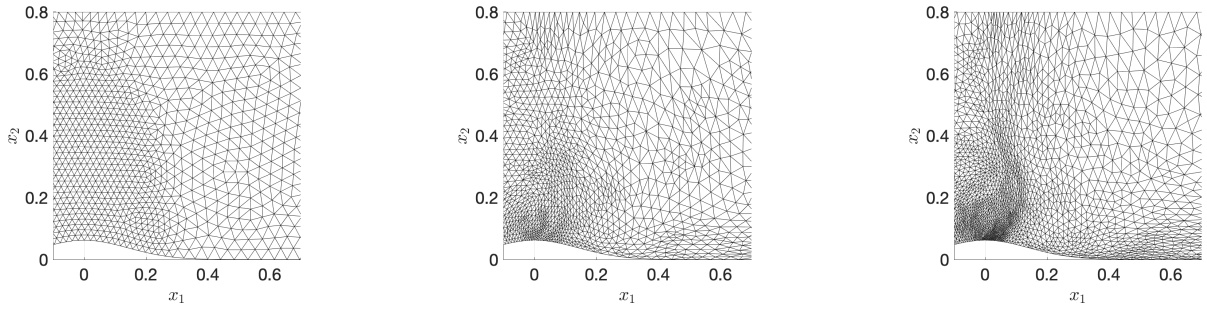


Figure 9: transonic bump. Performance of greedy sampling based on datasets of different fidelity over the test set.

7.2.3 Accelerated training with anisotropic mesh adaptation

We execute three iterations of Algorithm 1 with anisotropic mesh adaptation (cf. section 3). We initially set the parameter N in (24) equal to 750 and we increase it at each iteration by a factor 1.5. Figure 10 shows the sequence of meshes generated by Algorithm 1. We observe that the meshes are nearly isotropic in the proximity of the shock while they exhibit elongated elements in the downstream region (the minimum radius ratio is roughly 0.05). We notice that the adapted mesh for a single field is significantly more anisotropic in the proximity of the shock, but it becomes less and less anisotropic as we combine metrics associated with different parameters. This is likely due to the fact that the shock is not sharply tracked in the reference configuration.



(a) $it = 1$ (b) $it = 2$ (c) $it = 3$

Figure 10: transonic bump. Visualization of the reference mesh in the proximity of the bump, for three iterations of the adaptive algorithm (accelerated) with anisotropic mesh adaptation ($N_e = 3448$, $N_e = 4440$, $N_e = 6389$).

Table 4 compares performance of the accelerated training strategy based on isotropic and anisotropic MA: we observe that the two approaches lead to comparable performance for this model problem. We notice that the HF model requires slightly more PTC iterations to converge for anisotropic meshes: the difference is much more significant when we initialize the solver with the free-stream flow. This observation shows the importance of exploiting prior information to properly initialize the HF solver.

	L^2 error (avg)			enthalpy error (avg)		
	1	2	3	1	2	3
Isotropic MA	$0.29 \cdot 10^{-3}$	$0.31 \cdot 10^{-3}$	$0.49 \cdot 10^{-3}$	$0.59 \cdot 10^{-3}$	$0.37 \cdot 10^{-3}$	$0.23 \cdot 10^{-3}$
Anisotropic MA	$0.29 \cdot 10^{-3}$	$0.36 \cdot 10^{-3}$	$0.50 \cdot 10^{-3}$	$0.59 \cdot 10^{-3}$	$0.36 \cdot 10^{-3}$	$0.26 \cdot 10^{-3}$

	ROB size			online cost (avg)			offline cost
	1	2	3	1	2	3	
Isotropic MA	18	17	21	0.25	0.25	0.36	02:11:47
Anisotropic MA	18	17	16	0.26	0.26	0.24	01:58:48

Table 4: comparison of the performance of the accelerated adaptive procedures with isotropic and anisotropic mesh adaptation.

8 Summary and discussion

We developed and numerically validated an adaptive strategy for the simultaneous construction of high-fidelity and reduced-order approximations for parametric problems with discontinuous solutions. The approach relies on registration to track moving features of the solution field, metric-based mesh adaptation to devise an accurate mesh for the solution over a range of parameters, and projection-based model reduction to effectively estimate the (mapped) solution field. We show that registration is key to improve the compressibility of the solution manifold (cf. Figure 5) and enables parsimonious yet accurate HF approximations by complementing parameter-independent h-adaptation with parameter-dependent r-adaptation (cf. Figures 5 and 8). We also show that our adaptive training strategy provides increasingly more accurate approximations of the solution field (cf. Figures 4 and 7) and can be significantly accelerated by exploiting information from previous iterations (cf. section 6 and Tables 3 and 4).

We plan to extend our work in several directions. First, we wish to apply our framework to a broad range of problems in nonlinear mechanics, viscous compressible flows, and hydraulics: towards this end, we should extend our approach to unsteady PDEs and we should devise effective mesh and registration sensors for a broad range of solution features of interest. Second, we plan to leverage clustering techniques to further improve the quality of the HF mesh: even if MA allows us to optimize the size of the mesh, the HF meshes can still be large, in particular in the presence of parameter-induced topology changes that cannot be captured by a single parametric deformation; by resorting to clustering techniques, we hence expect to better control the distribution of the degrees of freedom in the spatio-parametric space.

Acknowledgements

The authors acknowledge the support provided by Inria through the exploratory action program (project title: *Adaptive Meshes for Model Order Reduction*, AM2OR). TT acknowledges the support by European Union's Horizon 2020 research and innovation program under the Marie Skłodowska-Curie Actions, grant agreement 872442 (ARIA).

A Further greedy procedures employed at training stage

In this section, we provide two greedy algorithms that are used during the execution of Algorithm 1. Algorithm 4 summarizes the parametric registration procedure for the transonic bump test case; on the other hand, Algorithm 5 outlines the strong greedy procedure employed to initialize Algorithm 2 and to select a subset of relevant solutions for mesh adaptation (cf. section 3).

We use notation

$$\left[\widehat{\mathbf{a}}_\mu, \mathfrak{f}^\star_\mu\right] \;=\; \texttt{registration}\left(s^{\mathrm{hf}}_\mu,\, \mathcal{S}_n,\, \mathtt{W}_{\mathrm{p}},\, \mathcal{T}_{\mathrm{hf}},\, \Psi^{\mathrm{geo}}_\mu,\, \mathbf{a}^0_\mu\right)$$

to refer to the function that takes as inputs (i) the target sensor $s^{\mathrm{hf}}_\mu : \Omega_{\mathrm{p}} \to \mathbb{R}$, (ii) the template space $\mathcal{S}_n$, (iii) the ROB $\mathtt{W}_m$ associated with the mapping space $\mathcal{U}_{\mathrm{p}} \subset \mathcal{U}_{\mathrm{hf,p}}$, (iv) the HF mesh $\mathcal{T}_{\mathrm{hf}}$, (v) the geometric mapping $\Psi^{\mathrm{geo}}_\mu : \Omega_{\mathrm{p}} \to \Omega_\mu$ and (vi) the initial guess $\mathbf{a}^0_\mu \in \mathbb{R}^m$ for the optimizer, and returns (I) the mapping coefficients $\widehat{\mathbf{a}}_\mu$ associated with a local minimum of the problem (28), and (II) the value of the target function $\mathfrak{f}^\star_\mu = \mathfrak{f}^{\mathrm{tg}}_\mu(\widehat{\mathbf{a}}_\mu)$. We also introduce the function

$$[\mathtt{W}_m,\; \{\mathbf{a}^{\mathrm{proj}}_\mu\}_{\mu\in\mathcal{P}_{\mathrm{train}}}] = \texttt{POD}\left(\{\widehat{\mathbf{a}}_\mu\}_{\mu\in\mathcal{P}_{\mathrm{train}}}, tol_{\mathrm{pod}}, (\cdot,\cdot)_2\right),$$

which implements POD based on the method of snapshots with Euclidean inner product $(\cdot,\cdot)_2$: the tolerance $tol_{\mathrm{pod}} > 0$ drives the selection of the number of modes m based on the energy criterion

$$m := \min\left\{ m' : \sum_{j=1}^{m'} \lambda_j \geq (1 - tol_{\mathrm{pod}}) \sum_{i=1}^{n_{\mathrm{train}}} \lambda_i \right\}, \tag{48}$$

where $\lambda_1 \geq \ldots \geq \lambda_{n_{\mathrm{train}}} \geq 0$ are the eigenvalues of the Gramian matrix $\mathbf{C} \in \mathbb{R}^{n_{\mathrm{train}} \times n_{\mathrm{train}}}$ such that $(\mathbf{C})_{k,k'} = \mathbf{a}^\star_{\mu^k} \cdot \mathbf{a}^\star_{\mu^{k'}}$. The function POD returns also the mapping coefficients associated with the projected displacements $\mathbf{a}^{\mathrm{proj}}_\mu$ onto the POD space; the latter are used to initialize the iterative method for the optimization problem in the subsequent iterations.

Algorithm 4 : registration algorithm ([48]).

Inputs: $\{s_\mu : \mu \in \mathcal{P}_{\text{train}}\}$ snapshot set, $\mathcal{S}_{n_0} = \text{span}\{s^{\text{hf}}_{\mu^{\star,(i)}}\}_{i=1}^{n_0}$ initial template space; $\mathcal{T}_{\text{hf}}$ mesh for HF computations.
Outputs: $\mathcal{S}_n$ template space, $\mathtt{W}_m : \mathbb{R}^m \to \mathcal{U}_{\text{p}}$ mapping ROB, $\{\varphi^\star_{\text{p},\mu^{\text{k}}} = \mathtt{W}_m \mathbf{a}^\star_{\mu^k}\}_k$ optimal mappings.

Initialization: $\mathcal{S}_{n=n_0} = \mathcal{S}_{n_0}$, $\Xi_\star = \{\mu^{\star,(i)}\}_{i=1}^{n_0}$, $\mathcal{U}_{\text{p}} = \mathcal{U}_{\text{hf,p}}$.
for $n = n_0, \dots, n_{\max} - 1$ **do**
  $\left[\widehat{\mathbf{a}}_\mu, \mathfrak{f}^\star_\mu\right] = \mathtt{registration}\left(s^{\text{hf}}_\mu, \mathcal{S}_n, \mathtt{W}_{\text{p}}, \mathcal{T}_{\text{hf}}, \mathbf{a}^0_\mu\right)$ for all $\mu \in \mathcal{P}_{\text{train}}$,
  see sections 4.3 and 6 for definition of $\mathbf{a}^0_\mu$
  $[\mathtt{W}_m,\ \{\mathbf{a}^{\text{proj}}_\mu\}_{\mu\in\mathcal{P}_{\text{train}}}] = \mathtt{POD}\left(\{\widehat{\mathbf{a}}_\mu\}_{\mu\in\mathcal{P}_{\text{train}}}, tol_{\text{pod}}, (\cdot,\cdot)_2\right)$,
  if $\max_{\mu\in\mathcal{P}_{\text{train}}} \mathfrak{f}^\star_\mu < \mathtt{tol}$ **then**, `break`
  else
    $\Xi_\star = \Xi_\star \cup \{\mu^{\star,(n+1)}\}$ with $\mu^{\star,(n+1)} = \arg\max_{\mu\in\mathcal{P}_{\text{train}}} \mathfrak{f}^\star_\mu$.
    $\mathcal{S}_{n+1} = \text{span}\{s^{\text{hf}}_{\mu^{\text{i},\star}} \circ \Phi_{\text{p},\mu^{\text{i},\star}}\}_{i=1}^{n+1}$.
  end if
end for

Algorithm 5 : strong-greedy algorithm (see, e.g., [40, section 7.3]).

Inputs: $\{\widehat{\boldsymbol{\alpha}}_\mu : \mu \in \mathcal{P}_{\text{train}}\} \subset \mathbb{R}^n$ snapshot set, $n_0 \le n$ size of the desired reduced space.
Outputs: $\mathcal{P}_\star = \{\mu^{\star,i}\}_{i=1}^{n_0}$ selected parameters.

Choose $\mathcal{Z} = \emptyset$, $\mathcal{P}_\star = \emptyset$.
for $i = 1, \dots, n_0$ **do**
  Compute $\mu^{\star,i} = \arg\max_{\mu\in\mathcal{P}_{\text{train}}} \min_{\boldsymbol{\alpha}\in\mathcal{Z}} \|\boldsymbol{\alpha} - \widehat{\boldsymbol{\alpha}}_\mu\|_2$
  Update $\mathcal{Z} = \mathcal{Z} \cup \text{span}\{\widehat{\boldsymbol{\alpha}}_{\mu^{\star,i}}\}$ and $\mathcal{P}_\star = \mathcal{P}_\star \cup \{\mu^{\star,i}\}$
end for

B Further numerical results for the transonic bump

In this section, we provide detailed results of the accelerated iterative procedure discussed in section 6. We distinguish between results obtained using isotropic and anisotropic mesh adaptation.

B.1 Acceleration with isotropic mesh adaptation

Figures 11, 12 and 13 replicate the results of Figures 6, 7 and 8: we observe that the results of the accelerated procedure are consistent with the ones obtained using the basic approach.

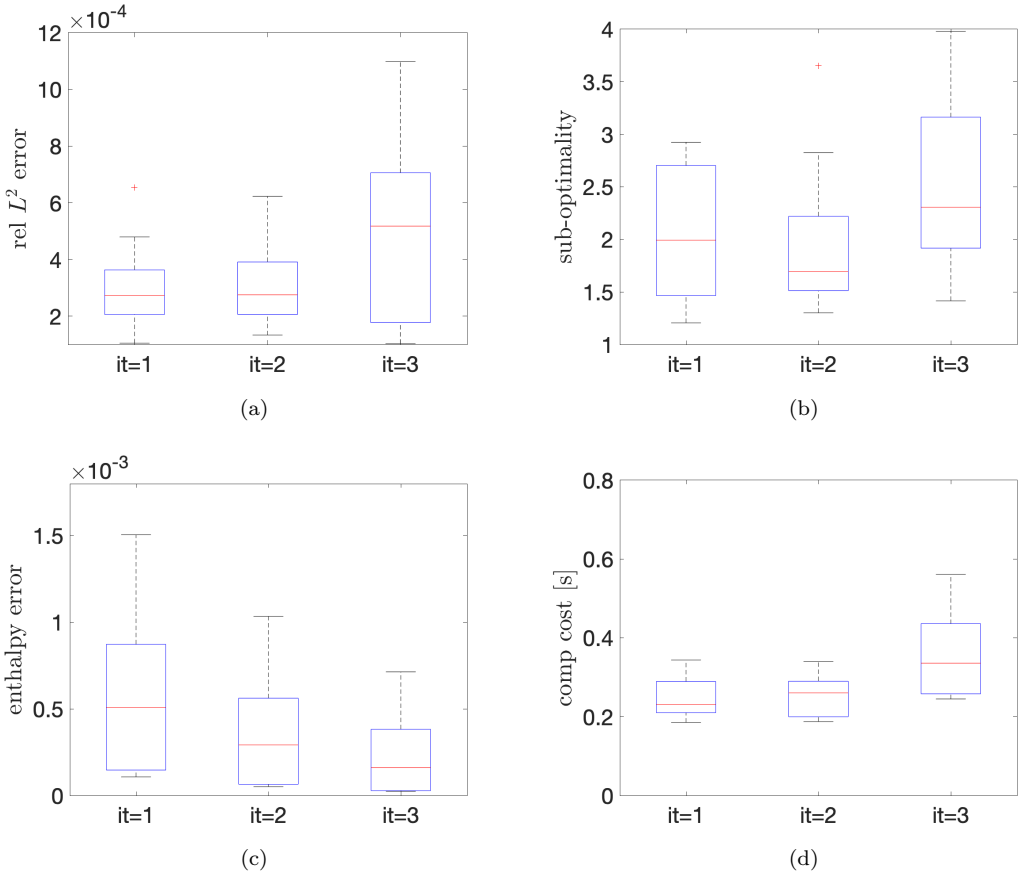


Figure 11: transonic bump. Performance of the ROM for three iterations of the adaptive (accelerated) procedure with isotropic mesh adaptation.

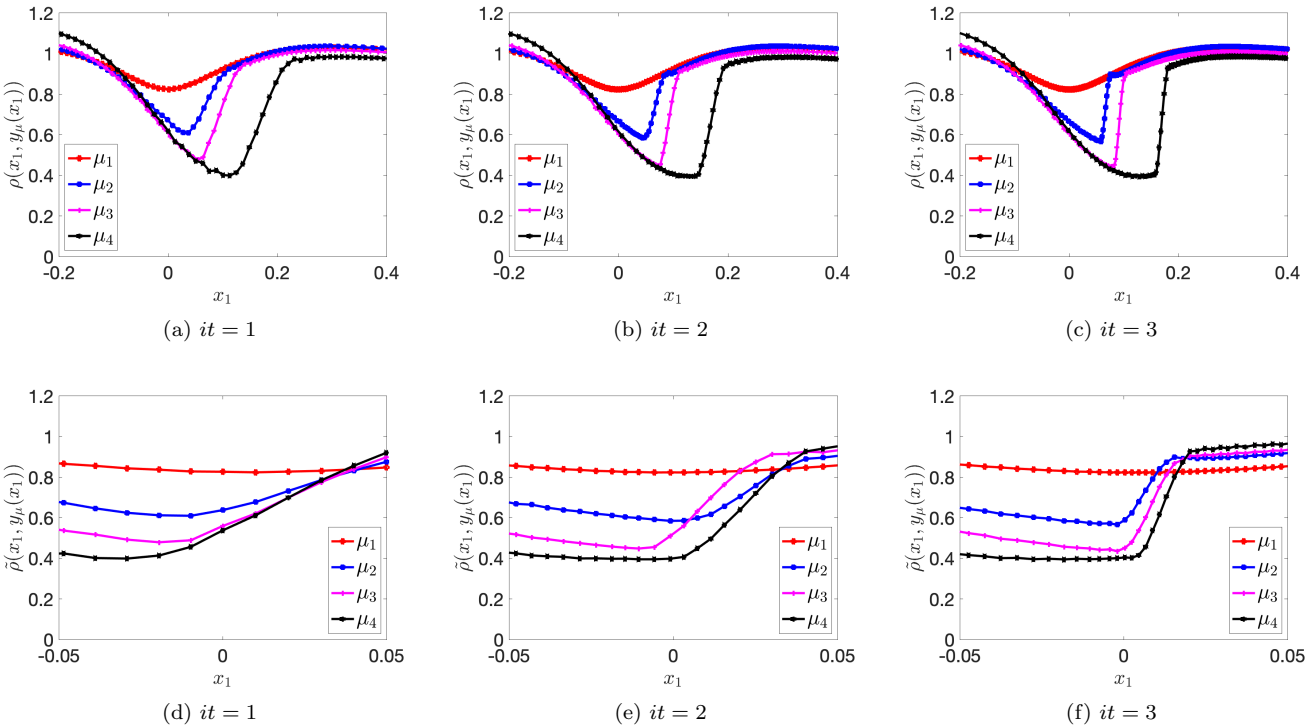
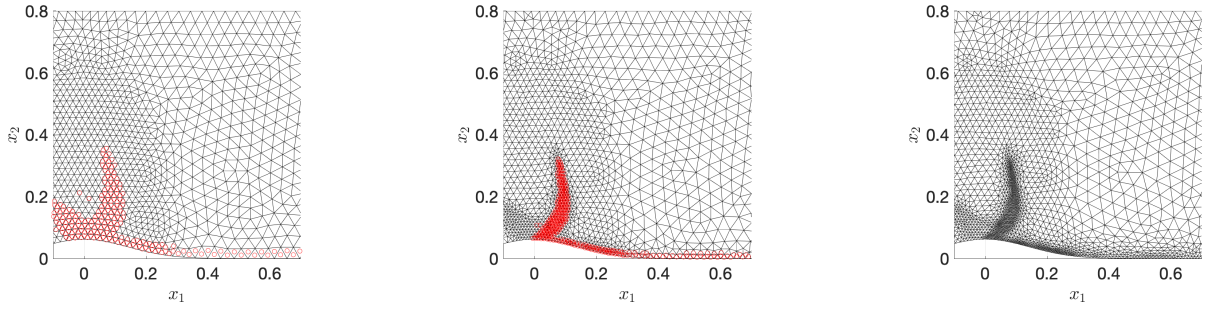


Figure 12: transonic bump. Behavior of the (modified) density field in physical (cf. (a)-(b)-(c)) and reference (cf. (d)-(e)-(f)) configuration for four values of the parameter and three iterations of the adaptive algorithm (accelerated version) with isotropic mesh adaptation.



(a) $it = 1$ (b) $it = 2$ (c) $it = 3$

Figure 13: transonic bump. Visualization of the reference mesh in the proximity of the bump, for three iterations of the adaptive algorithm (accelerated). Red dots indicate the centers of the marked elements.

Table 5 shows the details of the offline training costs. We observe that the vast majority of the computational gain is due to the reduction in the costs of the HF solves and also in the overhead of the greedy algorithm. We further remark that the acceleration strategy enables a much more efficient parallelization of the offline stage. First, the computation of the initial set of HF solutions (cf. Line 1, Algorithm 2) is embarrassingly parallel; second, the solution to the registration problems (34) based on the proposed initialization method is also parallel.

	it = 1	it = 2	it = 3
ROB size:	18	17	21
mesh size:	3448	4659	6663
snapshot generation:	1455.19	51.90	54.51
registration (sensor def.):	195.00	187.80	212.40
registration (optimization):	390.30	448.93	728.93
mesh adaptation:	0.00	0.36	0.61
greedy alg (HF solves):	460.26	725.28	1433.38
greedy alg (overhead):	282.49	284.42	994.63
PTC iterations (avg):	8.28	8.71	8.62

Table 5: transonic bump. Offline training costs (in seconds) of the adaptive (accelerated) approach.

B.2 Acceleration with anisotropic mesh adaptation

Figure 14 shows the performance of the ROM on the test set for the adaptive training procedure with anisotropic mesh adaptation: results are in good agreement with the results obtained using isotropic mesh adaptation. Similarly, Table 6 details the offline costs. As discussed in the main body of the paper, anisotropic MA leads to a slight increase in the number of PTC iterations required for convergence. Note, however, that the number of iterations is still much lower than the one obtained with free-stream solution initialization.

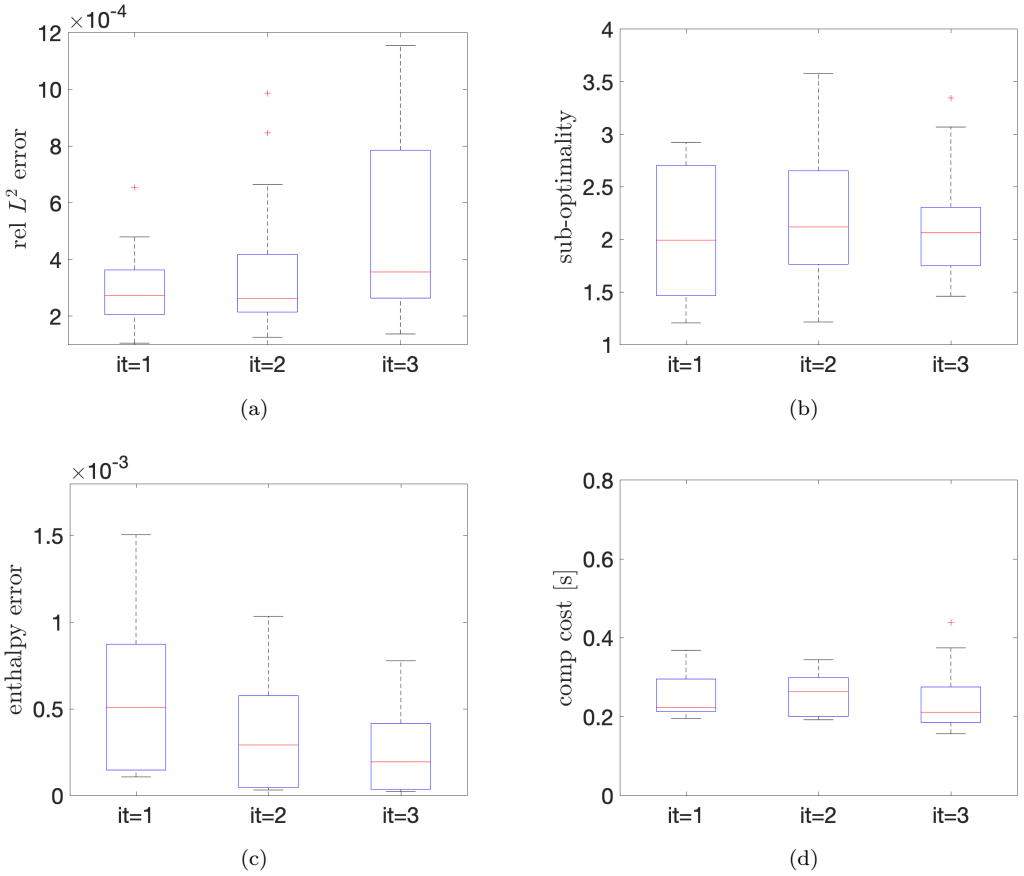


Figure 14: transonic bump. Performance of the ROM for three iterations of the adaptive (accelerated) procedure with anisotropic mesh adaptation.

	it = 1	it = 2	it = 3
ROB size:	18	17	16
mesh size:	3448	4440	6389
snapshot generation:	1426.79	51.47	55.77
registration (sensor def.):	177.64	186.56	191.43
registration (optimization):	388.84	442.69	692.53
mesh adaptation:	0.00	1.01	1.64
greedy alg (HF solves):	459.32	910.38	1357.74
greedy alg (overhead):	282.83	270.21	228.78
PTC iterations (avg):	8.28	11.65	10.19

Table 6: transonic bump. Offline training costs (in seconds) of the adaptive (accelerated) approach with anisotropic mesh adaptation.

References

[1] D. Amsallem and C. Farhat. Interpolation method for adapting reduced-order models and application to aeroelasticity. *AIAA journal*, 46(7):1803–1813, 2008.

[2] L. Arpaia, H. Beaugendre, L. Cirrottola, A. Froehly, M. Lorini, L. Nouveau, and M. Ricchiuto. H-and r-adaptation on simplicial meshes using MMG tools. In *Mesh Generation and Adaptation: Cutting-Edge Techniques*, pages 183–208. Springer, 2022.

[3] 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, 2022.

[4] N. Barral. *Time-accurate anisotropic mesh adaptation for three-dimensional moving mesh problems*. PhD thesis, Université Paris 6 Pierre et Marie Curie, 2015.

[5] N. Barral, G. Olivier, and F. Alauzet. Time-accurate anisotropic mesh adaptation for three-dimensional time-dependent problems with body-fitted moving geometries. *Journal of Computational Physics*, 331:157–187, 2017.

[6] F. Bassi and S. Rebay. A high-order accurate discontinuous finite element method for the numerical solution of the compressible Navier–Stokes equations. *Journal of computational physics*, 131(2):267–279, 1997.

[7] F. Black, P. Schulze, and B. Unger. Efficient wildland fire simulation via nonlinear model order reduction. *Fluids*, 6(8):280, 2021.

[8] H. Borouchaki, P. L. George, F. Hecht, P. Laug, and E. Saltel. Delaunay mesh generation governed by metric specifications. Part I. Algorithms. *Finite elements in analysis and design*, 25(1-2):61–83, 1997.

[9] C. J. Budd, W. Huang, and R. D. Russell. Adaptivity with moving grids. *Acta Numerica*, 18:111–241, 2009.

[10] K. Carlberg, M. Barone, and H. Antil. Galerkin v. least-squares Petrov–Galerkin projection in nonlinear model reduction. *Journal of Computational Physics*, 330:693–734, 2017.

[11] K. Carlberg, C. Farhat, J. Cortial, and D. Amsallem. The GNAT method for nonlinear model reduction: effective implementation and application to computational fluid dynamics and turbulent flows. *Journal of Computational Physics*, 242:623–647, 2013.

[12] T. Chapman, P. Avery, P. Collins, and C. Farhat. Accelerated mesh sampling for the hyper reduction of nonlinear computational models. *International Journal for Numerical Methods in Engineering*, 109(12):1623–1654, 2017.

[13] 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*, page 1247, 2022.

[14] P. Clément. Approximation by finite element functions using local regularization. *Revue Française d'Automatique, Informatique et Recherche Opérationnelle*, R-2:77–84, 1975.

[15] N. D. Dal Santo and A. Manzoni. Hyper-reduced order models for parametrized unsteady Navier-Stokes equations on domains with variable shape. *Advances in Computational Mathematics*, 45(5-6):2463–2501, 2019.

[16] C. Dapogny, C. Dobrzynski, and P. Frey. Three-dimensional adaptive domain remeshing, implicit domain meshing, and applications to free and moving boundary problems. *Journal of computational physics*, 262:358–378, 2014.

[17] 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*, page 111399, 2022.

[18] C. Farhat, T. Chapman, and P. Avery. Structure-preserving, stability, and accuracy properties of the energy-conserving sampling and weighting method for the hyper reduction of nonlinear finite element dynamic models. *International journal for numerical methods in engineering*, 102(5):1077–1110, 2015.

[19] L. Feng, L. Lombardi, G. Antonini, and P. Benner. Accelerating greedy algorithm for model reduction of complex systems by multi-fidelity error estimation. *arXiv preprint arXiv:2301.05610*, 2023.

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

[21] K. J. Fidkowski. *A high-order discontinuous Galerkin multigrid solver for aerodynamic applications*. PhD thesis, Massachusetts Institute of Technology, 2004.

[22] W. Huang and R. D. Russell. *Adaptive moving mesh methods*, volume 174. Springer Science & Business Media, 2010.

[23] A. Iollo and D. Lombardi. Advection modes by optimal mass transfer. *Physical Review E*, 89(2):022923, 2014.

[24] 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, 2022.

[25] M. Kast, M. Guo, and J. S. Hesthaven. A non-intrusive multifidelity method for the reduced order modeling of nonlinear problems. *Computer Methods in Applied Mechanics and Engineering*, 364:112947, 2020.

[26] P. Krah, S. Büchholz, M. Häringer, and J. Reiss. Front transport reduction for complex moving fronts. *Journal of scientific computing*, 96(28), 2023.

[27] 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.

[28] C. Little and C. Farhat. Nonlinear projection-based model order reduction in the presence of adaptive mesh refinement. In *AIAA SCITECH 2023 Forum*, page 2682, 2023.

[29] A. Loseille and F. Alauzet. Continuous mesh framework part I: well-posed continuous interpolation error. *SIAM Journal on Numerical Analysis*, 49(1):38–60, 2011.

[30] A. Loseille and F. Alauzet. Continuous mesh framework part II: validations and applications. *SIAM Journal on Numerical Analysis*, 49(1):61–86, 2011.

[31] MATLAB. *R2022a*. The MathWorks Inc., Natick, Massachusetts, 2022.

[32] A. T. McRae, C. J. Cotter, and C. J. Budd. Optimal-transport–based mesh adaptivity on the plane and sphere using finite elements. *SIAM Journal on Scientific Computing*, 40(2):A1121–A1148, 2018.

[33] 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, 2023.

[34] R. Mojgani and M. Balajewicz. Arbitrary Lagrangian Eulerian framework for efficient projection-based reduction of convection dominated nonlinear flows. In *APS Division of Fluid Dynamics Meeting Abstracts*, pages M1–008, 2017.

[35] 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, 2013.

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

[37] B. W.-L. Ong. *A new hr hybrid moving mesh-level set method*. PhD thesis, Simon Fraser University, 2007.

[38] B. Peherstorfer. Model reduction for transport-dominated problems via online adaptive bases and adaptive sampling. *SIAM Journal on Scientific Computing*, 42(5):A2803–A2836, 2020.

[39] P.-O. Persson and J. Peraire. Sub-cell shock capturing for discontinuous Galerkin methods. In *44th AIAA aerospace sciences meeting and exhibit*, page 112, 2006.

[40] A. Quarteroni, A. Manzoni, and F. Negri. *Reduced basis methods for partial differential equations: an introduction*, volume 92. Springer, 2015.

[41] L. Sirovich. Turbulence and the dynamics of coherent structures. I. Coherent structures. *Quarterly of applied mathematics*, 45(3):561–571, 1987.

[42] M. K. Sleeman and M. Yano. Goal-oriented model reduction for parametrized time-dependent nonlinear partial differential equations. *Computer Methods in Applied Mechanics and Engineering*, 388:114206, 2022.

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

[44] T. Taddei. An optimization-based registration approach to geometry reduction. *arXiv preprint arXiv:2211.10275*, 2022.

[45] T. Taddei, S. Perotto, and A. Quarteroni. Reduced basis techniques for nonlinear conservation laws. *ESAIM: Mathematical Modelling and Numerical Analysis*, 49(3):787–814, 2015.

[46] 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, 2021.

[47] T. Taddei and L. Zhang. Registration-based model reduction in complex two-dimensional geometries. *Journal of Scientific Computing*, 88(3):79, 2021.

[48] T. Taddei and L. Zhang. Space-time registration-based model reduction of parameterized one-dimensional hyperbolic PDEs. *ESAIM: M2AN*, 55(1):99–130, 2021.

[49] E. F. Toro. *Riemann solvers and numerical methods for fluid dynamics: a practical introduction*. Springer Science & Business Media, 2013.

[50] K. Veroy, C. Prud'Homme, D. Rovas, and A. Patera. A posteriori error bounds for reduced-basis approximation of parametrized noncoercive and nonlinear elliptic partial differential equations. In *16th AIAA Computational Fluid Dynamics Conference*, page 3847, 2003.

[51] S. Volkwein. Model reduction using proper orthogonal decomposition. *Lecture Notes, Institute of Mathematics and Scientific Computing, University of Graz. see http://www. uni-graz. at/imawww/volkwein/POD. pdf*, 1025, 2011.

[52] K. M. Washabaugh, M. J. Zahr, and C. Farhat. On the use of discrete nonlinear reduced-order models for the prediction of steady-state flows past parametrically deformed complex geometries. In *54th AIAA Aerospace Sciences Meeting*, page 1814, 2016.

[53] M. Yano. A reduced basis method for coercive equations with an exact solution certificate and spatio-parameter adaptivity: energy-norm and output error bounds. *SIAM Journal on Scientific Computing*, 40(1):A388–A420, 2018.

[54] M. Yano. Discontinuous Galerkin reduced basis empirical quadrature procedure for model reduction of parametrized nonlinear conservation laws. *Advances in Computational Mathematics*, 45(5):2287–2320, 2019.

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

[56] M. Yano, J. Modisette, and D. Darmofal. The importance of mesh adaptation for higher-order discretizations of aerodynamic flows. In *20th AIAA Computational Fluid Dynamics Conference*, page 3852, 2011.

[57] J. Yu and J. S. Hesthaven. A study of several artificial viscosity models within the discontinuous galerkin framework. *Communications in Computational Physics*, 27(5):1309–1343, 2020.

[58] M. J. Zahr, A. Shi, and P.-O. Persson. Implicit shock tracking using an optimization-based high-order discontinuous Galerkin method. *Journal of Computational Physics*, 410:109385, 2020.