

Friedrichs' systems discretized with the Discontinuous Galerkin method: domain decomposable model order reduction and Graph Neural Networks approximating vanishing viscosity solutions

Francesco Romor*, Davide Torlo† and Gianluigi Rozza‡

Mathematics Area, mathLab, SISSA, via Bonomea 265, I-34136 Trieste, Italy

August 8, 2023

Abstract

Friedrichs' systems (FS) are symmetric positive linear systems of first-order partial differential equations (PDEs), which provide a unified framework for describing various elliptic, parabolic and hyperbolic semi-linear PDEs such as the linearized Euler equations of gas dynamics, the equations of compressible linear elasticity and the Dirac-Klein-Gordon system. FS were studied to approximate PDEs of mixed elliptic and hyperbolic type in the same domain. For this and other reasons, the versatility of the discontinuous Galerkin method (DGM) represents the best approximation space for FS. We implement a distributed memory solver for stationary FS in `deal.II`. Our focus is model order reduction. Since FS model hyperbolic PDEs, they often suffer from a slow Kolmogorov n-width decay. We develop two approaches to tackle this problem. The first is domain decomposable reduced-order models (DD-ROMs). We will show that the DGM offers a natural formulation of DD-ROMs, in particular regarding interface penalties, compared to the continuous finite element method. We also develop new repartitioning strategies to obtain more efficient local approximations of the solution manifold. The second approach involves graph neural networks used to infer the limit of a succession of projection-based linear ROMs corresponding to lower viscosity constants: the heuristic behind is to develop a multi-fidelity super-resolution paradigm to mimic the mathematical convergence to vanishing viscosity solutions while exploiting the most interpretable and certified projection-based ROMs.

1 Introduction

Friedrichs' systems (FS) are a class of symmetric positive linear systems of first-order partial derivative equations (PDEs). They were introduced by Friedrichs [34] as a tool to study hyperbolic and elliptic phenomena in different parts of the domain within a unifying framework. The main ideas that allow recasting many models into the FS frameworks are the introduction of extra variables to lower the order of the higher derivatives and the linearization of nonlinear problems. FS are characterized by linear and positive operators and (non-uniquely defined) boundary operators that allow them to impose classical boundary conditions (BCs). Various works proved uniqueness, existence and well-posedness of the FS in their strong, weak and ultraweak formulation and the necessary conditions to properly define the boundary operators [34, 72, 73, 28, 31, 3, 24].

In the last decades, different numerical discretizations of the FS have been proposed to approximate the analytical solutions. The strategies vary among finite volume [78] and discontinuous Galerkin (DG) formulations [45, 52, 28, 29, 31, 30, 12, 17]. Along with the DG discretization, also error estimation analysis that provide, according to the type of edge penalization, optimal or sub-optimal estimates, have been carried out [28, 29, 24]. We focus on the DG method since it is more versatile to approximate both elliptic and hyperbolic PDEs and it fits naturally in the framework of domain decomposable ROMs (DD-ROMs).

In the context of parametric PDEs, for multi-query tasks or real-time simulations, fast and reliable simulations of the same problem for different parameters are often needed. This is especially true when the full-order models (FOMs) are based on expensive and high-order DG discretizations. Reduced order models (ROMs) decrease the computational costs looking for the solutions of unseen parametric instances on low-dimensional discretization spaces, called reduced basis spaces. This is possible because the new solutions to be predicted are expected to be highly correlated with the database of training DG solutions used to build the reduced spaces. ROMs have been proven to be a powerful tool for many applications

*francesco.romor@sissa.it

†davide.torlo@sissa.it

‡gianluigi.rozza@sissa.it

[68, 61, 44, 76]. In particular for linear problems, classical Galerkin and Petrov-Galerkin projection methods are very easy to set up and extremely convenient in terms of computational costs. FS are perfectly suited for such algorithms due to their linearity. This is a preliminary step needed to reduce parametric nonlinear PDEs whose linearization results in FS. In the most simple formulation, we will apply singular value decomposition (SVD) to compress a database of snapshots and provide a reliable reduced order model (ROM), with standard *a posteriori* error estimators.

In the context of model order reduction, FS are particularly beneficial as theoretical frameworks for many reasons. They represent a new form of structure-preserving ROMs: the positive symmetric properties of FS are in fact easily inherited by the reduced numerical formulations. This advocates for the employment of FS for reduced order modelling whenever a PDE can be reformulated in the FS framework. This is the case for the Euler equations of gas dynamics, when they are written in terms of entropy variables [78, 65]. The same rationale is behind structure-preserving symplectic or Hamiltonian ROMs [43] and port-Hamiltonian ROMs [83, 10]. Moreover, since FS are often studied in their ultraweak formulation, they are good candidates for optimally stable error estimates [13] at the full-order level [12], also in a hybridized DG implementation in [17], and at the reduced order level, similarly to what has been achieved in the works [11, 39, 42]. Finally, from the point of view of software design, the possibility to implement in a unique maintainable and generic manner the realization of ROMs for PDEs ascribed to the class of FS is a convenient feature to search for.

Though being linear, FS are hyperbolic systems and often show an advection dominated character, which is not easily approximable through a simple proper orthogonal decomposition (POD). This leads to a slow Kolmogorov n -width (KnW) decay that results in very inefficient approximations of the reduced models. Several approaches have been studied to overcome this difficulty [79, 49, 66, 74, 16, 15, 2, 59, 20, 81, 51].

A strategy that has been developed to reduce PDEs solved numerically with domain decomposition approaches, like fluid-structure interaction systems, are domain decomposable ROMs (DD-ROMs). The initial formulations [61, 62, 48, 27, 48] involved continuous finite elements discretizations for which new ways to couple the solutions restricted to different subdomains needed to be devised, especially to enforce continuity at the interfaces. We show that the DGM imposes naturally flux interface penalties from the full-order discretizations and it is, thus, amenable for straightforward implementations of DD-ROMs. From the point of view of solution manifold approximability and so KnW decay, DD-ROMs are based on local linear approximants that are employed to reach a higher accuracy for unseen solutions. This is useful when the computational domain is divided in subdomains that are independently affected by the parametric instances. The typical case in which this may happen are parametric models for which discontinuous values of the parameters over fixed subdomains cause non correlated responses on their respective subdomains. Similar cases will be studied in sections 5.3.1 and 5.3.2. Another example is represented by parametric fluid-structure interaction systems in which the parameters cause complex interdependencies between the structure and fluid components in favor of partitioned linear solution manifold approximations (SVD is performed separately for the fluid, for the structure and for the interface) rather than monolithic ones. In our implementation of DD-ROMs, we exploit the partitions obtained from the distributed memory solver in `deal.II`. Since these domain decompositions typically satisfy constraints related to the computational efficiency, we devise some strategies to repartition the domain responding to solution manifold approximability concerns instead. Another work that implements this is [86] where the Reynolds stress tensor is employed, among others, as indicator for partitioning the computational domain. Similarly, we develop new indicators.

Another way to approach the problem of a slow KnW decay is exploiting the mathematical proofs of existence of vanishing viscosity solutions [64, 57, 25, 38]. In fact, solutions of hyperbolic problems can be obtained as a limit process of solutions associated to viscosity terms approaching zero. The crucial point is that ROMs associated to larger viscosity values may not suffer from a slow Kolmogorov n -width decay. Hence, we can set up classical projection based ROMs for the high viscosity solutions, and use graph neural networks (GNNs) [80] only to infer the vanishing viscosity solution in a very efficient manner. This procedure can be applied also to more general hyperbolic problems, not necessarily FS. The key features of this new methodology are the following: the employment of computationally heavy graph neural networks is reduced to a minimum and, at the same time, interpretable certified projection ROMs are exploited as much as possible in their regime of accurate linear approximability. In fact, GNNs, used generally to perform non-intrusive MOR, have high training computational costs and they are employed mainly for small academic benchmarks in terms of number of degrees of freedom, up to now. We avoid these high computational efforts with our multi-fidelity formulation: the GNNs are employed only to infer the vanishing viscosity solutions from the previous higher viscosity level, not to approximate and perform dimension reduction of the entire solution manifold. The overhead is the collection of additional full-order snapshots corresponding to high viscosity values, but this can be performed on coarser meshes as it will be done in section 6. Moreover, the support of our GNNs is the DG discretization space, so, we can enrich the typical machine learning framework of GNNs with data structure and operators from numerical analysis. We validate the use of data augmentation with numerical filters (discretized Laplacian, gradients), as proposed in [80].

In brief, we summarize our contributions with the present work:

- structure-preserving model order reduction for Friedrichs' systems. We synthetically describe the realization of ROMs for FS and the definition of standard *a posteriori* error estimators. Hints towards the implementation of optimally stable ROMs are highlighted.

- domain decomposable reduced-order models for full-order models discretized with the discontinuous Galerkin method. We introduce DD-ROMs for DG discretizations and introduce novel indicators to repartition the computational domain with the aim of obtaining more efficient local solution manifold approximants.
- surrogate modelling of vanishing viscosity solutions with graph neural networks. We propose a new framework for the MOR of parametric hyperbolic PDEs with a slow Kolmogorov n -width decay.

The topics addressed in this work are presented as follows. In Section 2, we introduce the definition of FS and well-posedness results and we will provide several examples of models that fall into this framework: the Maxwell equations in stationary regime, the equations of linear compressible elasticity and the advection diffusion reaction equations. Then, we provide a DG discretization of the FS following [24] with related error estimates in Section 3. In Section 4, we introduce the projection-based MOR technique and some error bounds that can be effectively used. In Section 5, we will discuss a new implementation of domain decomposable ROMs for FOMs discretized with the DGM and we will test the approach on three parametric models. In Section 6, we introduce the concept of vanishing viscosity solutions and how graph neural network are exploited to overcome the problem of a slow Kolmogorov n -width decay. We will provide some numerical tests to show the effectiveness of the proposed approach. Finally, in Section 7 we summarize our results and we suggest further directions of research.

2 Friedrichs' systems

In this section, we will provide a summary of FS theory: their definition, existence, uniqueness and well-posedness results, their weak and ultraweak forms and many PDEs which can be rewritten into FS. The following discussion collects many results from [34, 72, 47, 73, 46, 52, 78, 52, 29, 31, 30, 3, 12, 24], but we will follow the notation in [24]. Let us represent with d the ambient space dimension and with $m \geq 1$ the number of equations of the FS. We consider a connected Lipschitz domain $\Omega \subset \mathbb{R}^d$, with boundary $\partial\Omega$ and outward unit normal $\mathbf{n} : \partial\Omega \rightarrow \mathbb{R}^d$.

A FS is defined through $(d+1)$ matrix-valued fields $A^0, A^1, \dots, A^d \in [L^\infty(\Omega)]^{m \times m}$ and the following differential operators $\mathcal{X}, A, \tilde{A} : \Omega \rightarrow \mathbb{R}^{m \times m}$. We suppose that $\mathcal{X} \in [L^\infty(\Omega)]^{m \times m}$ and define

$$\mathcal{X} = \sum_{k=1}^d \partial_k A^k, \quad A = A^0 + \sum_{i=1}^d A^i \partial_i, \quad \tilde{A} = (A^0)^T - \mathcal{X} - \sum_{i=1}^d A^i \partial_i, \quad (1)$$

assuming that

$$A^k = (A^k)^T \text{ a.e. in } \Omega, \quad \text{for } k = 1, \dots, d, \quad (\text{symmetry property}) \quad (2a)$$

$$A^0 + (A^0)^T - \mathcal{X} \text{ is u.p.d. a.e. in } \Omega, \quad (\text{positivity property}) \quad (2b)$$

thus, the name **symmetric positive operators** or **Friedrichs operators**, which is used to refer to (A, \tilde{A}) . We recall that the operator in (2b) is uniformly positive definite (u.p.d) if and only if

$$\exists \mu > 0 : A^0 + (A^0)^T - \mathcal{X} > 2\mu_0 \mathbb{I} \quad \text{a.e. in } \Omega. \quad (3)$$

If this property is not satisfied, it can be sometimes recovered as shown in Appendix A. A weaker condition can be required for two-field systems [30].

The boundary conditions are expressed through two boundary operators $\mathcal{D} : \partial\Omega \rightarrow \mathbb{R}^{m \times m}$ with

$$\mathcal{D} = \sum_{k=1}^d n_k A^k, \quad \text{a.e. in } \partial\Omega \quad (4)$$

and $\mathcal{M} : \partial\Omega \rightarrow \mathbb{R}^{m \times m}$ satisfying the following **admissible boundary conditions**

$$\mathcal{M} \text{ is nonnegative a.e. on } \partial\Omega, \quad (\text{monotonicity property}) \quad (5a)$$

$$\ker(\mathcal{D} - \mathcal{M}) + \ker(\mathcal{D} + \mathcal{M}) = \mathbb{R}^m \text{ a.e. on } \partial\Omega. \quad (\text{strict adjointness property}) \quad (5b)$$

Remark 1 (Strict adjointness). The term strict adjointness property comes from Jensen [52, Theorem 31]. The strict adjointness property is needed for the solution of the ultra-weak formulation of the FS to uniquely satisfy the boundary conditions: in a slightly different framework from the one presented here, see [52, Theorem 29] and [12, proof of Lemma 2.4].

Theorem 1 (Friedrichs' system strong solution [34]). Let $f \in [L^2(\Omega)]^m$, the strong solution $z \in [C^1(\bar{\Omega})]^m$ to Friedrichs' system

$$\begin{cases} Az = f, & \text{in } \Omega, \\ (\mathcal{D} - \mathcal{M})z = 0, & \text{on } \partial\Omega. \end{cases} \quad (6)$$

is unique. Moreover, there exists a solution of the ultra-weak formulation

$$(z, \tilde{A}y)_{L^2} = (f, y)_{L^2}, \quad \forall y \in [C^1(\bar{\Omega})]^m \text{ s.t. } (\mathcal{D} + \mathcal{M}^t)y = 0. \quad (7)$$

Let $L = [L^2(\Omega)]^m$. We define the weak formulation on the graph space $V = \{z \in L : Az \in L\}$, which amounts to differentiability in the characteristics directions: $A \in \mathcal{L}(V, L)$ and $\tilde{A} \in \mathcal{L}(V', L)$. The boundary operator \mathcal{D} is translated into the abstract operator $D \in \mathcal{L}(V, V')$:

$$\langle Dz, y \rangle_{V, V'} = \langle Az, y \rangle_L - (z, \tilde{A}y)_L, \quad \forall z, y \in V. \quad (8)$$

When z is smooth, it can be seen as the integration by parts formula [52, 12]:

$$\langle Dz, y \rangle_{V, V'} = \langle \mathcal{D}z, y \rangle_{H^{\frac{1}{2}}(\partial\Omega), H^{-\frac{1}{2}}(\partial\Omega)}, \quad \forall z \in H^1(\Omega), y \in H^1(\Omega). \quad (9)$$

A sufficient condition for well-posedness of the weak formulation is provided by the cone formalism [3, 24] that poses the existence of two linear subdomains (V_0, V_0^*) of V :

$$V_0 \text{ maximal in } C^+, \quad V_0^* \text{ maximal in } C^- \quad (10a)$$

$$V_0 = D(V_0^*)^\perp, \quad V_0^* = D(V_0)^\perp, \quad (10b)$$

such that $A : V_0 \rightarrow L$ and $\tilde{A} : V_0^* \rightarrow L$ are isomorphisms, where $C^\pm = \{w \in V \mid \pm \langle Dw, w \rangle_{V, V'} \geq 0\}$.

Provided that $V_0 + V_0^* \subset V$ is closed [3], the conditions in (10) are equivalent to the existence of the boundary operator $M \in \mathcal{L}(V, V')$ that satisfies **admissible boundary conditions** analogue to the ones in (5):

$$M \text{ is monotone,} \quad (\text{monotonicity property}) \quad (11a)$$

$$\ker(D - M) + \ker(D + M) = V, \quad (\text{strict adjointness property}) \quad (11b)$$

identifying $V_0 = \ker(D - M)$ and $V_0^* = \ker(D + M^*)$.

Theorem 2 (Friedrichs' System weak form [28, 29, 24, 12]). Let us assume that the boundary operator $M \in \mathcal{L}(V, V')$ satisfies the monotonicity and strict adjointness properties (11). Let us define for $z, z^* \in V$ the bilinear forms

$$a(z, y) = (Az, y)_L + \frac{1}{2}\langle (D - M)z, y \rangle_{V', V}, \quad \forall y \in V, \quad (12a)$$

$$a^*(z^*, y) = (\tilde{A}z^*, y)_L + \frac{1}{2}\langle (D + M^*)z^*, y \rangle_{V', V}, \quad \forall y \in V. \quad (12b)$$

Then, Friedrichs' operators $A : V_0 \rightarrow L$ and $\tilde{A} : V_0^* \rightarrow L$ are isomorphisms: for all $f \in L$ and $g \in V$ there exists unique $z, z^* \in V$ s.t.

$$a(z, y) = (f, y)_L + \langle (D - M)g, y \rangle_{V', V} \quad \forall y \in V, \quad (13a)$$

$$a^*(z^*, y) = (f, y)_L + \langle (D + M^*)g, y \rangle_{V', V} \quad \forall y \in V, \quad (13b)$$

that is

$$\begin{cases} Az = f, & \text{in } L, \\ (M - D)(z - g) = 0 & \text{in } V', \end{cases} \quad \begin{cases} \tilde{A}z^* = f, & \text{in } L, \\ (M^* + D)(z^* - g) = 0 & \text{in } V'. \end{cases} \quad (14)$$

2.1 A unifying framework

The theory of Friedrichs' systems provides a unified framework to study different classes of PDEs [52]: first-order uniformly hyperbolic, second-order uniformly hyperbolic, elliptic and parabolic partial differential equations. Originally, Friedrichs' aim was to study equations of mixed type (hyperbolic, parabolic, elliptic) inside the same domain such as the Tricomi equation [34] (or more generally the Frankl equation [52]) inspired by models from compressible gas dynamics for which the domain is subdivided in a hyperbolic supersonic and an elliptic subsonic part.

Some examples of FS can be found in the literature:

$$(x_2 \partial_1^2 \bullet + \partial_1^2 \bullet) u = 0, \quad (\text{Tricomi [34]}) \quad (15a)$$

$$\begin{bmatrix} -\partial_1 \bullet & \partial_2 \bullet \\ -\partial_2 \bullet & \partial_1 \bullet \end{bmatrix} \begin{pmatrix} u_1 \\ u_2 \end{pmatrix} = 0, \quad (\text{Cauchy-Riemann [34]}) \quad (15b)$$

$$(A(x_2)\partial_1^2 + \partial_1^2)u = 0, \quad (\text{Frankl [52]}) \quad (15c)$$

$$\begin{bmatrix} \mathbb{I}_3 & -\lambda^{-1}\mathbb{I}_3(\nabla \cdot \bullet) - \frac{(\nabla \bullet + (\nabla \bullet)^t)}{2} \\ -\frac{1}{2}\nabla \cdot (\bullet + \bullet^t) & \alpha\mathbb{I}_3 \end{bmatrix} \begin{pmatrix} \sigma \\ \mathbf{u} \end{pmatrix} = 0, \quad (\text{Compressible linear elasticity [28]}) \quad (15d)$$

$$\begin{bmatrix} \mu\mathbb{I}_3 & \nabla \times \bullet \\ -\nabla \times \bullet & \sigma\mathbb{I}_3 \end{bmatrix} \begin{pmatrix} \mathbf{H} \\ \mathbf{E} \end{pmatrix} = 0, \quad (\text{Maxwell eq. in stationary regime [28]}) \quad (15e)$$

$$(-\nabla \cdot (\kappa\nabla \bullet) + \beta \cdot \nabla \bullet + \mu \bullet) \mathbf{u} = 0, \quad (\text{Diffusion advection reaction [28]}) \quad (15f)$$

$$(A_0 \partial_t \bullet + \sum_{i=1}^3 \tilde{A}_i \partial_i \bullet) \mathbf{V} = 0, \quad (\text{Linearized symmetric Euler [47, 78]}) \quad (15g)$$

$$(a\gamma^0 \partial_t \bullet + \gamma^1 \partial_1 \bullet + \gamma^2 \partial_2 \bullet + \gamma^3 \partial_3 \bullet + B) \psi = 0, \quad (\text{Dirac system [4]}) \quad (15h)$$

$$\begin{bmatrix} -ai\gamma^0 \partial_t \bullet - i\gamma^1 \partial_1 \bullet - i\gamma^2 \partial_2 \bullet - i\gamma^3 \partial_3 \bullet + M & \mathbf{1}_4 \\ \mathbf{0}_4 & \partial_t^2 \bullet - \Delta \bullet + m^2 \mathbb{I}_1 \end{bmatrix} \begin{pmatrix} \psi \\ \phi \end{pmatrix} = \mathbf{f}, \quad (\text{Dirac-Klein-Gordon system [4]}) \quad (15i)$$

$$\begin{bmatrix} -\frac{i}{2\pi}(a\gamma^0 \partial_t \bullet + \gamma^1 \partial_1 \bullet + \gamma^2 \partial_2 \bullet + \gamma^3 \partial_3 \bullet + B) & \mathbf{1}_4 \\ \mathbf{0}_4 & -\partial_t^2 \bullet + \Delta \bullet \end{bmatrix} \begin{pmatrix} \psi \\ \mathcal{A} \end{pmatrix} = \mathbf{f}, \quad (\text{Maxwell-Dirac system [4]}) \quad (15j)$$

$$\begin{bmatrix} -i\omega\mu\mathbb{I}_3 & \nabla \times \bullet \\ -\nabla \times \bullet & (-i\omega\epsilon + \sigma)\mathbb{I}_3 \end{bmatrix} \begin{pmatrix} \mathbf{H} \\ \mathbf{E} \end{pmatrix} = 0, \quad (\text{Time-harmonic Maxwell [4]}) \quad (15k)$$

$$\begin{bmatrix} \nu\mathbb{I}_3 & \nabla \times \bullet \\ \mu\beta \times \bullet - \nabla \times \bullet & \sigma\mathbb{I}_3 \end{bmatrix} \begin{pmatrix} \mathbf{H} \\ \mathbf{E} \end{pmatrix} = \mathbf{f}, \quad (\text{Magneto-hydrodynamics [29]}) \quad (15l)$$

$$\begin{bmatrix} \nu^{-1}\mathbb{I}_3 & \mathbf{1}_3 & -\frac{(\nabla \bullet + (\nabla \bullet)^t)}{2} \\ \text{tr}(\bullet) & d\mathbb{I}_1 & 0 \\ -\frac{1}{2}\nabla \cdot (\bullet + \bullet^t) & 0 & \beta \cdot \nabla \bullet \end{bmatrix} \begin{pmatrix} \sigma \\ p \\ \mathbf{u} \end{pmatrix} = \mathbf{f}, \quad (\text{Incompressible linearized Navier-Stokes [30]}) \quad (15m)$$

for the employed notation we refer to the respective reported references. A non-stationary version of (15e) and (15f) from [24] is omitted. The FS framework here presented easily extends to complex-valued systems as in (15h), (15i), (15j) and (15k) from [4]. We will consider only semi-linear PDEs but FS can be encountered as intermediate steps when solving quasi-linear PDEs: for example solving the compressible Euler equations of gas dynamics in entropy variables with the Newton method brings to the FS (15g), as studied in [78].

One of the critical points of FS is the definition of the boundary conditions. Friedrichs's idea [34] was to impose boundary conditions through a matrix-valued boundary operator. Ern, Guermond and Caplain [31] revised the FS theory, without employing the trace of functions in the graph space as developed in [72, 73, 46, 52], but in terms of operators acting in abstract Hilbert spaces, as presented here.

The most common homogeneous boundary conditions (homogeneous Dirichlet, Neumann, Robin) for (15d), (15e) and (15f) can be found in the literature [28, 29, 31]. For a choice of boundary conditions, and thus for a choice of spaces (V_0, V_0^*) , there can be more than one definition of the boundary operator M [31, Remark 5.3]. A constructive methodology for defining the boundary operator $M \in \mathcal{L}(V, V')$ from specific boundary conditions can be found in [31] and it will be employed for the compressible linear elasticity test case in Section 2.1.2. Also inhomogeneous boundary conditions can be imposed through the definition of traces of functions in graph spaces as in [52] or through a Petrov-Galerkin formulation as in [12]. In the following, we will present in detail three FS on which we will focus in the numerical test section.

2.1.1 Curl–curl problem: Maxwell equations in stationary regime

We will consider the Maxwell equations in stationary regime, also known as the curl–curl problem. Let $\mathbf{E} \in \mathbb{R}^d = \mathbb{R}^3$ be the electric field and $\mathbf{H} \in \mathbb{R}^3$ be the magnetic field. The curl–curl problem is defined as

$$\begin{cases} \mu\mathbf{H} + \nabla \times \mathbf{E} = \mathbf{r}, \\ \sigma\mathbf{E} - \nabla \times \mathbf{H} = \mathbf{g}, \end{cases}$$

with $\mu, \sigma > 0$, the permeability and permittivity constants. The FS is obtained by setting

$$A^0 = \begin{bmatrix} \mu\mathbb{I}_{d,d} & 0_{d,d} \\ 0_{d,d} & \sigma\mathbb{I}_{d,d} \end{bmatrix}, \quad A^k = \begin{bmatrix} 0_{d,d} & \mathcal{R}^k \\ (\mathcal{R}^k)^T & 0_{d,d} \end{bmatrix}, \quad f = \begin{pmatrix} \mathbf{r} \\ \mathbf{g} \end{pmatrix}$$

with $\mathcal{R}_{ij}^k = \epsilon_{ijk}$ being the Levi-Civita tensor. The graph space is $V = H(\text{curl}, \Omega) \times H(\text{curl}, \Omega)$. The boundary operator is

$$\mathcal{D} = \sum_{k=1}^d n_k A^k = \begin{bmatrix} 0_{d,d} & \mathcal{T} \\ \mathcal{T}^T & 0_{d,d} \end{bmatrix}, \quad \text{with } \mathcal{T}\xi := \mathbf{n} \times \boldsymbol{\xi}, \quad (16)$$

$$\langle D(\mathbf{H}, \mathbf{E}), (\mathbf{h}, \mathbf{e}) \rangle_{V', V} = (\mathbf{n} \times \mathbf{E}, \mathbf{e})_{L^2(\partial\Omega)} - (\mathbf{n} \times \mathbf{H}, \mathbf{h})_{L^2(\partial\Omega)}. \quad (17)$$

We impose homogeneous Dirichlet boundary conditions tangential to the electric field $(\mathbf{n} \times \mathbf{E})|_{\partial\Omega} = 0$ through

$$\mathcal{M} = \begin{bmatrix} 0_{d,d} & -\mathcal{T} \\ \mathcal{T}^T & 0_{d,d} \end{bmatrix}, \quad \langle M(\mathbf{H}, \mathbf{E}), (\mathbf{h}, \mathbf{e}) \rangle_{V', V} = -(\mathbf{n} \times \mathbf{E}, \mathbf{e})_{L^2(\partial\Omega)} - (\mathbf{n} \times \mathbf{H}, \mathbf{h})_{L^2(\partial\Omega)}. \quad (18)$$

2.1.2 Compressible linear elasticity

We consider the parametric compressible linear elasticity system in $\mathbb{R}^d = \mathbb{R}^3$, where $\boldsymbol{\sigma} \in \mathbb{R}^{d \times d}$ is the stress tensor and $\mathbf{u} \in \mathbb{R}^d$ is the displacement vector. The system can be written as

$$\begin{pmatrix} \boldsymbol{\sigma} - \mu_1(\nabla \cdot \mathbf{u})\mathbb{I}_{3,3} - 2\mu_2 \frac{(\nabla \mathbf{u} + (\nabla \mathbf{u})^t)}{2} \\ -\frac{1}{2}\nabla \cdot (\boldsymbol{\sigma} + \boldsymbol{\sigma}^t) + \mu_3 \mathbf{u} \end{pmatrix} = \begin{pmatrix} \mathbf{0} \\ \mathbf{r} \end{pmatrix}, \quad \forall x \in \Omega, \quad (19)$$

where $\mathbf{r} \in \mathbb{R}^3$, and $\mu_1, \mu_2 > 0$ are the Lamé constants. Rescaling the displacement \mathbf{u} by $2\mu_2$, we obtain

$$\begin{pmatrix} \boldsymbol{\sigma} - \frac{\mu_1}{2\mu_2+3\mu_1} \text{tr}(\boldsymbol{\sigma})\mathbb{I}_{3,3} - \frac{(\nabla \mathbf{u} + (\nabla \mathbf{u})^t)}{2} \\ -\frac{1}{2}\nabla \cdot (\boldsymbol{\sigma} + \boldsymbol{\sigma}^T) + \frac{\mu_3}{2\mu_2} \mathbf{u} \end{pmatrix} = \begin{pmatrix} \mathbf{0} \\ \mathbf{r} \end{pmatrix}, \quad \forall x \in \Omega. \quad (20)$$

In this case, we consider the graph space

$$V = H_{\boldsymbol{\sigma}} \times [H^1(\Omega)]^d, \quad H_{\boldsymbol{\sigma}} = \{\boldsymbol{\sigma} \in [L^2(\Omega)]^{d \times d} \mid \nabla \cdot (\boldsymbol{\sigma} + \boldsymbol{\sigma}^t) \in [L^2(\Omega)]^d\}. \quad (21)$$

If we reorder the coefficients of $\boldsymbol{\sigma}$ into a vector, we can define $\mathbf{z} = \begin{pmatrix} \boldsymbol{\sigma} \\ \mathbf{u} \end{pmatrix}$ and have

$$A^0 = \begin{bmatrix} \mathbb{I}_{d^2, d^2} - \frac{\mu_1}{2\mu_2+3\mu_1} \mathcal{Z} & 0_{d^2, d} \\ 0_{d, d^2} & \frac{\mu_3}{2\mu_2} \mathbb{I}_{d, d} \end{bmatrix}, \quad A^k = \begin{bmatrix} 0_{d^2, d^2} & \mathcal{E}^k \\ (\mathcal{E}^k)^T & 0_{d, d} \end{bmatrix}, \quad f = \begin{bmatrix} \mathbf{0}_{d^2} \\ \mathbf{r} \end{bmatrix}, \quad (22)$$

with $\mathcal{Z}_{[ij], [kl]} = \delta_{ij}\delta_{kl}$ and $\mathcal{E}_{[ij], l}^k = -\frac{1}{2}(\delta_{ik}\delta_{jl} + \delta_{il}\delta_{jk})$. This leads to the definition of the boundary operator

$$\mathcal{D} = \sum_{k=1}^d n_k A^k = \begin{bmatrix} 0_{d^2, d^2} & \mathcal{N} \\ \mathcal{N}^T & 0_{d, d} \end{bmatrix} \text{ with } \mathcal{N}\boldsymbol{\xi} := -\frac{1}{2}(\mathbf{n} \otimes \boldsymbol{\xi} + \boldsymbol{\xi} \otimes \mathbf{n}), \quad (23a)$$

$$\langle D(\boldsymbol{\sigma}, \mathbf{u}), (\boldsymbol{\tau}, \mathbf{v}) \rangle_{V', V} = -\langle \frac{1}{2}(\boldsymbol{\sigma} + \boldsymbol{\sigma}^t) \cdot \mathbf{n}, \mathbf{v} \rangle_{-\frac{1}{2}, \frac{1}{2}} - \langle \frac{1}{2}(\boldsymbol{\tau} + \boldsymbol{\tau}^t) \cdot \mathbf{n}, \mathbf{u} \rangle_{-\frac{1}{2}, \frac{1}{2}}. \quad (23b)$$

Mixed boundary conditions $\mathbf{u}|_{\Gamma_D} = 0$ and $(\boldsymbol{\sigma} \cdot \mathbf{n})|_{\Gamma_N} = 0$ can be applied through the following boundary operator on the Dirichlet boundary Γ_D and on the Neumann boundary Γ_N :

$$\begin{aligned} \langle M(\boldsymbol{\sigma}, \mathbf{u}), (\boldsymbol{\tau}, \mathbf{v}) \rangle_{V', V} = & -\langle \frac{1}{2}(\boldsymbol{\sigma} + \boldsymbol{\sigma}^t) \cdot \mathbf{n}, \mathbf{v} \rangle_{-\frac{1}{2}, \frac{1}{2}, \Gamma_D} + \langle \frac{1}{2}(\boldsymbol{\tau} + \boldsymbol{\tau}^t) \cdot \mathbf{n}, \mathbf{u} \rangle_{-\frac{1}{2}, \frac{1}{2}, \Gamma_D} \\ & + \langle \frac{1}{2}(\boldsymbol{\sigma} + \boldsymbol{\sigma}^t) \cdot \mathbf{n}, \mathbf{v} \rangle_{-\frac{1}{2}, \frac{1}{2}, \Gamma_N} - \langle \frac{1}{2}(\boldsymbol{\tau} + \boldsymbol{\tau}^t) \cdot \mathbf{n}, \mathbf{u} \rangle_{-\frac{1}{2}, \frac{1}{2}, \Gamma_N}, \end{aligned} \quad (24)$$

the constructive procedure employed to define the boundary operator $M \in \mathcal{L}(V, V')$ is reported in the Appendix B.

2.1.3 Grad–div problem: advection–diffusion–reaction equations

Another example is the advection–diffusion–reaction equation

$$-\nabla \cdot (\kappa \nabla u) + \boldsymbol{\beta} \cdot \nabla u + \mu u = \mathbf{r}, \quad (25)$$

with $\kappa \in [L^\infty(\Omega)]^{d \times d}$ and $\boldsymbol{\beta} \in [W^{1,\infty}(\Omega)]^d$, under the hypothesis that $\kappa \in [L^\infty(\Omega)]^{d \times d}$ and $\mu - \nabla \cdot \boldsymbol{\beta} \in L^\infty(\Omega)$ are uniformly bounded from below to satisfy the positivity property (2b). Let us write the equation in the mixed form with $\boldsymbol{\sigma} = -\kappa \nabla u$ and $\mathbf{z} = \begin{pmatrix} \boldsymbol{\sigma} \\ u \end{pmatrix}$. Then, (25) can be rewritten as (6) with

$$A^0 = \begin{bmatrix} \kappa^{-1} & 0_{d,1} \\ 0_{1,d} & \mu \end{bmatrix}, \quad A^k = \begin{bmatrix} 0_{d,d} & \mathbf{e}_k \\ (\mathbf{e}_k)^T & \beta_k \end{bmatrix}, \quad \mathbf{f} = \begin{pmatrix} 0 \\ \mathbf{r} \end{pmatrix}. \quad (26)$$

Here, $0_{m,\ell} \in \mathbb{R}^{m \times \ell}$ is a matrix of zeros and \mathbf{e}_k is the unitary vector with the k -th entry equal to 1. The graph space is $V = H(\text{div}, \Omega) \times H^2(\Omega)$. The boundary operator D becomes

$$D = \sum_{k=1}^d n_k A^k = \begin{bmatrix} 0_{d,d} & \mathbf{n} \\ \mathbf{n}^t & \beta \cdot \mathbf{n} \end{bmatrix}, \quad \langle D(\boldsymbol{\sigma}, u), (\boldsymbol{\tau}, v) \rangle_{V', V} = \langle \boldsymbol{\sigma} \cdot \mathbf{n}, v \rangle_{-\frac{1}{2}, \frac{1}{2}} - \langle \boldsymbol{\tau} \cdot \mathbf{n}, u \rangle_{-\frac{1}{2}, \frac{1}{2}}. \quad (27)$$

Homogeneous Dirichlet boundary conditions $u|_{\partial\Omega} = 0$ can be imposed with

$$\mathcal{M} = \begin{bmatrix} 0_{d,d} & -\mathbf{n} \\ \mathbf{n}^t & 0 \end{bmatrix}, \quad (28)$$

while Robin/Neumann boundary conditions of the type $\boldsymbol{\sigma} \cdot \mathbf{n} = \gamma u$ are imposed with

$$\mathcal{M} = \begin{bmatrix} 0_{d,d} & \mathbf{n} \\ -\mathbf{n}^t & 2\gamma + \beta \cdot \mathbf{n} \end{bmatrix}. \quad (29)$$

For our test case in Section 6, we will consider as advection field $\boldsymbol{\beta} : \Omega \rightarrow \mathbb{R}^d$ an incompressible velocity field from the solution of the 2d incompressible Navier-Stokes equations as described later. Similarly to the linear compressible elasticity mixed boundary conditions in Section 2.1.2, we want to impose $u|_{\Gamma_D} = \mathbf{g} \in [L^{\frac{1}{2}}(\Gamma_D)]^d$ and $(\boldsymbol{\sigma} \cdot \mathbf{n})|_{\Gamma_N} = 0$. This is possible with

$$\langle M(\boldsymbol{\sigma}, u), (\boldsymbol{\tau}, v) \rangle_{V', V} = \langle \boldsymbol{\sigma} \cdot \mathbf{n}, v \rangle_{-\frac{1}{2}, \frac{1}{2}, \Gamma_D} + \langle \boldsymbol{\tau} \cdot \mathbf{n}, u \rangle_{-\frac{1}{2}, \frac{1}{2}, \Gamma_D} - \langle \boldsymbol{\sigma} \cdot \mathbf{n}, v \rangle_{-\frac{1}{2}, \frac{1}{2}, \Gamma_N} - \langle \boldsymbol{\tau} \cdot \mathbf{n}, u \rangle_{-\frac{1}{2}, \frac{1}{2}, \Gamma_N}, \quad (30)$$

the proof is similar to the one reported in Appendix B.

3 Discontinuous Galerkin discretization

In the literature, a few discretization approaches for FS are presented, e.g. finite volume method [78] or discontinuous Galerkin (DG) method [28, 24, 12]. More recently, a hp-adaptive hybridizable DG formulation was introduced in [17]. In this work, we perform a DG discretization following the notation reported in [28, 24]. Consider a shape-regular tessellation \mathcal{T}_h of the domain Ω and take a piecewise polynomial space V_h over \mathcal{T}_h , defined by $V_h = \{z \in V : z|_T \in \mathbb{P}_d^k(T), \forall T \in \mathcal{T}_h\}$, where k is the polynomial degree. We assume that there is a partition $P_\Omega = \{\Omega_i\}_{1 \leq i \leq N_\Omega}$ of Ω into disjoint polyhedra such that the exact solution z belongs to $V^* = V \cap [H^1(P_\Omega)]^m$. We define the discrete bilinear form $\forall y_h \in V_h, z \in V^*$

$$a_h^{cf}(z, y_h) = \sum_{T \in \mathcal{T}_h} (Az, y_h)_{L^2(T)} + \frac{1}{2} \sum_{F \in \mathcal{F}_h^b} ((\mathcal{M} - \mathcal{D})z, y_h)_{L^2(F)} - \sum_{F \in \mathcal{F}_h^i} (\mathcal{D}_F [\![z]\!], \{\{y_h\}\})_{L^2(F)} \quad (31a)$$

$$= \sum_{T \in \mathcal{T}_h} (z, \tilde{A}y_h)_{L^2(T)} + \frac{1}{2} \sum_{F \in \mathcal{F}_h^b} ((\mathcal{M} + \mathcal{D})z, y_h)_{L^2(F)} + \sum_{F \in \mathcal{F}_h^i} (\mathcal{D}_F \{\{z\}\}, [\![y_h]\!])_{L^2(F)}, \quad (31b)$$

where the first two terms are the piece-wise discontinuous discretization of the bilinear form (12a) and the last term penalizes the jump across neighboring cells and stabilizes the method. Here, \mathcal{F}_h^b is the collection of the faces of the triangulation \mathcal{T}_h belonging to the boundary of Ω_h , while \mathcal{F}_h^i is the collection of internal faces. The jump and the average of a function on a face F shared by two elements T_1 and T_2 are defined as $[\![u]\!] = u|_{T_1} - u|_{T_2}$ and $\{\{u\}\} = \frac{1}{2}(u|_{T_1} + u|_{T_2})$, respectively. The boundary operator $\mathcal{D} : \partial\Omega \rightarrow \mathbb{R}^{m \times m}$ can be extended also on the internal faces $F \in \mathcal{F}_h^i$ as $\mathcal{D}_F = \sum_{k=1}^d n_k^F A^k$, where n^F is a normal to the face F and it is well-defined.

In order to obtain quasi-optimal error estimates, extra stabilization terms are needed. We additionally impose that $A^i \in [C^{0, \frac{1}{2}}(\overline{\Omega}_j)]^{m \times m}$, $\forall T \in \mathcal{T}_h$, $i = 1, \dots, d$, $\forall \Omega_j \in P_\Omega$. A possibility is given by the following stabilization term

$$s_h(z, y_h) = \sum_{F \in \mathcal{F}_h^b} (S_F^b z, y_h)_{L^2(F)} + \sum_{F \in \mathcal{F}_h^i} (S_h^i [\![z]\!], [\![y_h]\!])_{L^2(F)}, \quad (32)$$

where the operators S_h^i and S_F^b have to satisfy the following constraints for some $\alpha_j > 0$ for $j = 1, \dots, 5$:

$$S_F^b z = 0 \quad \forall F \in \mathcal{F}_h^b, \quad S_F^i [\![z]\!] = 0 \quad \forall F \in \mathcal{F}_h^i, \quad \text{with } z \text{ the exact solution,} \quad (33a)$$

$$S_F^b \text{ and } S_F^i \text{ are symmetric and nonnegative,} \quad (33b)$$

$$S_F^b \leq \alpha_1 \mathbb{I}_{m,m}, \quad \alpha_2 |D_F| \leq S_F^i \leq \alpha_3 \mathbb{I}_{m,m}, \quad (33c)$$

$$|(M - D)y, z\rangle_{L^2(F)} \leq \alpha_4 ((S_F^b + M)y, y)_{L^2(F)}^{1/2} \|z\|_{L^2(F)}, \quad (33d)$$

$$|(M + D)y, z\rangle_{L^2(F)} \leq \alpha_5 ((S_F^b + M)z, z)_{L^2(F)}^{1/2} \|y\|_{L^2(F)}. \quad (33e)$$

Specific definitions of these operators for our test cases are presented in [28, 24], properly declined for our mixed boundary conditions in the compressible linear elasticity and advection–diffusion–reaction test cases, see Sections 2.1.2 and 2.1.3, respectively. Finally, we can define the bilinear form and the right-hand side

$$a_h(z, y_h) = a_h^{cf}(z, y_h) + s_h(z, y_h), \quad l_h(y_h) = \sum_{T \in \mathcal{T}_h} (f, y_h)_{L^2(T)} + \frac{1}{2} \sum_{F \in \mathcal{F}_h^b} ((M - D)g, y_h)_{L^2(F)} \quad (34)$$

that lead to the definition of the discrete problem.

Definition 1 (DG Friedrichs' System). *Given $f \in L$ and $g \in V_h$, the DG approximation of the FS constitute in finding a $z_h \in V_h$ such that*

$$a_h(z_h, y_h) = l_h(y_h), \quad \forall y_h \in V_h. \quad (35)$$

To prove the accuracy of the discrete problem, it is necessary to have the following conditions:

- Consistency, i.e., $a_h(z, y_h) = a(z, y_h)$ for $z \in V^*$;
- L^2 -coercivity, i.e., $a_h(y_h, y_h) \geq \mu_0 \|y_h\|_L^2 + \frac{1}{2} |y_h|_M^2$, with $|y_h|_M^2 = \int_{\partial\Omega} y^t M y$;
- Inf-sup stability

$$\|z_h\| \lesssim \sup_{y_h \neq 0} \frac{a_h(z_h, y_h)}{\|y_h\|} \quad (36)$$

with $\|y\|^2 = \|y\|_{L^2}^2 + |y|_M^2 + |y|_S^2 + \sum_{T \in \mathcal{T}_h} h_T \|A^k \partial_k y\|_{L^2(T)}^2$ and $|y|_S^2 = s_h(y, y)$;

- Boundedness $a_h(w, y_h) \lesssim \|w\|_* \|y_h\|$ with

$$\|y\|_*^2 = \|y\|^2 + \sum_{T \in \mathcal{T}_h} \left(h_T^{-1} \|y\|_{L^2(T)}^2 + \|y\|_{L^2(\partial T)}^2 \right). \quad (37)$$

Theorem 3 (Error estimate from [28, 24]). *Let $z \in V^*$ be the solution of the weak problem (13a) and $z \in V_h$ be the solution of the discrete DG problem (35). Then, the consistency and inf-sup stability of the discrete system (35) imply*

$$\|z - z_h\| \lesssim \inf_{y_h \in V_h} \|z - y_h\|_*, \quad (38)$$

in particular, if $z \in [H^{k+1}(\Omega)]^m$ the following convergence rate holds

$$\|z - z_h\| \lesssim h^{k+\frac{1}{2}} \|z\|_{[H^{k+1}(\Omega)]^m}. \quad (39)$$

4 Projection-based model order reduction

The computation of discrete solutions of parametrized PDEs can require a not negligible computational time. In particular, in multi-query context, when many evaluations for different parameters are required, the computations may become unbearable. In this section, we introduce a reduced order model (ROM) for the FS in case of parameter dependent problems [44, 76], in order to drastically reduce the computational costs. To do so, we exploit two aspects of the above presented FS: the linearity of the problems and the affine dependence of the operators on the physical parameters.

As we have seen in Section 2.1, all the problems are depending on some parameters $\boldsymbol{\rho} \in \mathcal{P} \subset \mathbb{R}^{N_{\text{par}}}$ and the dependence is affine. This means that it is possible to find N_{aff} terms independent on the parameters for each form, such that they can be affinely combined with some parameter dependent functions to obtain the original operator, i.e.,

$$a_h(z, y_h; \boldsymbol{\rho}) = \sum_{\ell=1}^{N_{\text{aff}}} \theta_\ell^a(\boldsymbol{\rho}) a_{\ell,h}(z, y_h), \quad l_h(y_h) = \sum_{\ell=1}^{N_{\text{aff}}} \theta_\ell^f(\boldsymbol{\rho}) l_{h,l}(y_h). \quad (40)$$

Then, we select a reduced space $V_r \subset V_h$ provided by a compression algorithm, e.g. SVD/POD/PCA [53, 58, 82] or Greedy algorithm [69, 68, 44, 20]. We suppose that the reduced dimension r is much smaller than the dimension N_h of the full order model space V_h . We look as ansatz for a reduced solution $z_{\text{RB}} \in V_r$ a linear combination of the bases $\{\psi_j^{\text{RB}}\}_{j=1}^r$ of V_r , i.e.,

$$z_{\text{RB}} = \sum_{j=1}^r z_{\text{RB}}^j \psi_j^{\text{RB}}, \quad (41)$$

then, performing a standard Galerkin projection, we obtain the following RB problem.

Definition 2 (Reduced Basis Problem). *Find $z_{RB} \in V_r$, given by the coefficients z_{RB}^j , such that*

$$\sum_{i=1}^r z_{RB}^j(\boldsymbol{\rho}) \sum_{\ell=1}^{N_{\text{aff}}} \theta_\ell^a(\boldsymbol{\rho}) a_{\ell,h}(\psi_j^{RB}, \psi_i^{RB}) = \sum_{\ell=1}^{N_{\text{aff}}} \theta_\ell^f(\boldsymbol{\rho}) (f_\ell, \psi_i^{RB}), \quad \text{for all } i = 1, \dots, r. \quad (42)$$

The obtained problem scales depend on the dimension r and N_{aff} in its assembly and only on r in its solution, and it is completely independent on N_h . To obtain computational advantages for the parametric problem, we split the tasks into an expensive *offline phase* and a cheap *online phase*. In the *offline phase*, we find the reduced space V_r and we assemble the reduced matrices and right hand sides

$$A_\ell := \{a_{\ell,h}(\psi_j^{RB}, \psi_i^{RB})\}_{i,j}, \quad b_\ell := \{(f_\ell, \psi_i^{RB})\}_i. \quad (43)$$

In the *online phase*, we can simply evaluate the coefficients $\theta_\ell^a(\boldsymbol{\rho})$ and $\theta_\ell^f(\boldsymbol{\rho})$ and obtain the reduced linear system

$$A(\boldsymbol{\rho}) z_{RB} = b(\boldsymbol{\rho}), \quad \text{with } A(\boldsymbol{\rho}) := \sum_{\ell} \theta_\ell^a(\boldsymbol{\rho}) A_\ell \text{ and } b(\boldsymbol{\rho}) := \sum_{\ell} \theta_\ell^f(\boldsymbol{\rho}) b_\ell. \quad (44)$$

This gives a great speed up in computational times.

4.1 Reduced basis a posteriori error estimate

We derive two error estimators for the energy norm and the L^2 norm of the reduced basis error $e_h = z_h - z_{RB} \in V_h$ following the procedure in [44]. Exploiting the equality in (31), we obtain the following lower bound

$$a_h(e_h, e_h) = a_h^{cf}(e_h, e_h) + s_h(e_h, e_h) \quad (45a)$$

$$= \sum_{T \in \mathcal{T}_h} (Ae_h, e_h)_{L^2(T)} + \frac{1}{2} \sum_{F \in \mathcal{F}^b} ((\mathcal{M} - \mathcal{D})e_h, e_h)_{L^2(F)} - \sum_{F \in \mathcal{F}_h^i} (\mathcal{D}_F [\![e_h]\!], \{e_h\})_{L^2(F)} + \quad (45b)$$

$$\sum_{F \in \mathcal{F}_h^b} (S_F^b e_h, e_h)_{L^2(F)} + \sum_{F \in \mathcal{F}_h^i} (S_h^i [\![e_h]\!], [\![e_h]\!])_{L^2(F)} \quad (45c)$$

$$= \sum_{T \in \mathcal{T}_h} ((A^0 - \frac{1}{2}\mathcal{X})e_h, e_h)_{L^2(T)} + \frac{1}{2} \sum_{F \in \mathcal{F}^b} (\mathcal{M}e_h, e_h)_{L^2(F)} + \sum_{F \in \mathcal{F}_h^b} (S_F^b e_h, e_h)_{L^2(F)} + \sum_{F \in \mathcal{F}_h^i} (S_h^i [\![e_h]\!], [\![e_h]\!])_{L^2(F)} \quad (45d)$$

$$\geq \mu_0 \|e_h\|_L^2 + \frac{1}{2} |e_h|_M^2 + |e_h|_S^2, \quad (45e)$$

where we have defined $|\cdot|_M^2 = \frac{1}{2} \sum_{F \in \mathcal{F}^b} (\mathcal{M} \cdot, \cdot)_{L^2(F)}$ and $|\cdot|_S^2 = s_h(\cdot, \cdot)$. We define the R -norm

$$\|y_h\|_R^2 = \mu_0 \|y_h\|_L^2 + \frac{1}{2} |y_h|_M^2 + |y_h|_S^2, \quad \forall y_h \in V_h, \quad (46)$$

that may depend on ρ only through μ_0 and is generated by the scalar product

$$\langle u_h, v_h \rangle_R = \mu_0 \sum_{T \in \mathcal{T}_h} (u_h, v_h)_{L^2(T)} + \frac{1}{2} \sum_{F \in \mathcal{F}^b} (\mathcal{M}^{sym} u_h, v_h)_{L^2(F)} + \sum_{F^b} (S_F^b u_h, v_h) + \sum_{F^i} (S_h^i u_h, v_h). \quad (47)$$

The boundary operators we will employ in our benchmarks are all skew-symmetric so $\mathcal{M}^{sym} = \frac{\mathcal{M} + \mathcal{M}^t}{2}$ is the null matrix and $|e_h|_M = 0$. Now, we can proceed to provide an *a posteriori* error estimate for the R -norm and energy norm. Hence, let us define $r_{RB}(y_h) = l_h(y_h; \rho) - a_h(z_{RB}, y_h; \rho)$ and its R and L -Riesz representations as \hat{r}_R and \hat{r}_L such that

$$r_{RB}(u_h) = \langle \hat{r}_R, u_h \rangle_R, \quad X_R \hat{r}_R = \mathbf{L}_h - A_h \mathbf{z}_{RB}, \quad r_{RB}(u_h) = \langle \hat{r}_L, u_h \rangle_L, \quad X_L \hat{r}_L = \mathbf{L}_h - A_h \mathbf{z}_{RB}, \quad (48)$$

where X_R and X_L are the R -norm and L -norm mass matrices, and \mathbf{L}_h , A_h and \mathbf{z}_{RB} are the representations of $l_h(\cdot; \rho)$, $a_h(\cdot, \cdot; \rho)$ and z_{RB} in the DG basis of V_h . The \hat{r}_L representation can be computed cheaply when the parametric model is affinely decomposable with respect to the parameters, while \hat{r}_R requires the inversion of a possibly parametric dependent matrix X_R .

Now, consider the energy norm of the error $\|e_h\|_{nrg}^2 = a_h(e_h, e_h)$ and the coercivity constant $\|e_h\|_{nrg}^2 \geq \mu_0 \|e_h\|_L^2$ derived in (45e), we have the following *a posteriori* error estimates

$$\frac{\|e_h\|_{nrg}}{\|z_h\|_{nrg}} \leq \frac{\|\hat{r}_R\|_R}{\|z_h\|_{nrg}}, \quad \frac{\|e_h\|_R}{\|z_h\|_R} \leq \frac{\|\hat{r}_R\|_R}{\|z_h\|_R}, \quad \frac{\|e_h\|_{nrg}}{\|z_h\|_{nrg}} \leq \frac{\|\hat{r}_L\|_L}{\sqrt{\mu_0} \|z_h\|_{nrg}}, \quad \frac{\|e_h\|_L}{\|z_h\|_L} \leq \frac{\|\hat{r}_L\|_L}{\mu_0 \|z_h\|_L}, \quad (49)$$

namely, the relative energy error with the corresponding *a posteriori* R -norm energy estimate, the relative R -norm error with the corresponding *a posteriori* R -norm estimate, the relative energy error with the corresponding *a posteriori* L -norm energy estimate, and the relative L -norm error with the corresponding *a posteriori* L -norm estimate.

4.2 Optimally stable error estimates for the ultraweak Petrov-Galerkin formulation

In this section, we show that Friedrichs' systems are a desirable unifying formulation to consider when performing model order reduction also due to the possibility to achieve an optimally stable formulation. This can further simplify the error estimator analysis reaching the equality between the error and the residual norm. This is not the first case in which optimally stable formulations are introduced also at the reduced level, see [11, 39, 42]. In the following, we describe how to achieve this ultraweak formulation and we delineate the path one should follow to use such formulation. Nevertheless, we will not use this formulation in our numerical tests and we leave the implementation to future studies.

We introduce the following Discontinuous Petrov-Galerkin (DPG) formulation from [12]. To do so, we first define $V(\mathcal{T}_h)$ the broken graph space with norm $\|\bullet\|_{V(\mathcal{T}_h)}^2 = \|\bullet\|_L^2 + \sum_{T \in \mathcal{T}_h} \|A\bullet\|_{L^2(T)}^2$ and $\tilde{V} = V/Q(\Omega)$ the quotient of the graph space V with

$$\begin{aligned} Q(\Omega) &= \left\{ z \in V \mid \sum_{T \in \mathcal{T}_h} \langle Dz, y \rangle_{\tilde{V}(T), V(T)} + \frac{1}{2} \langle (M - D)z, y \rangle_{\tilde{V}(\Omega), V(\Omega)} = 0, \quad \forall y \in V(\mathcal{T}_h) \right\} \\ &= \left\{ z \in V \mid a(z, y) = 0, \quad \forall y \in V(\mathcal{T}_h) \right\}. \end{aligned} \quad (50)$$

The DPG formulation reads: find $(z, q) \in L \times \tilde{V}$ such that, for all $y \in V(\mathcal{T}_h)$,

$$\sum_{T \in \mathcal{T}_h} (z, \tilde{A}y)_{L^2(T)} + \sum_{T \in \mathcal{T}_h} \langle Dq, y \rangle_{\tilde{V}(T), V(T)} + \frac{1}{2} \langle (M - D)q, y \rangle_{\tilde{V}, V} = \sum_{T \in \mathcal{T}_h} (f, v)_{\tilde{V}(T), V(T)} + \frac{1}{2} \langle (M - D)g, y \rangle_{\tilde{V}, V}. \quad (51)$$

The introduction of the hybrid face variables $q \in \tilde{V}$ is necessary since $z \in L$ does not satisfy (8). In practice, assuming that the traces of $y \in V(\mathcal{T}_h)$ are well-defined and belong to a space $X(\mathcal{F}_{i,b})$, we can formulate (51) as follows: find $(z, q) \in L \times X(\mathcal{F}_{i,b})$ such that, for all $y \in V(\mathcal{T}_h)$,

$$\sum_{T \in \mathcal{T}_h} (z, \tilde{A}y)_{L^2(T)} + \sum_{F \in \mathcal{F}_i} (\mathcal{D}q, [\![y]\!])_{X(F)} + \frac{1}{2} \sum_{F \in \mathcal{F}_b} ((\mathcal{M} - \mathcal{D})q, y)_{X(F)} = \sum_{T \in \mathcal{T}_h} (f, v)_{\tilde{V}(T), V(T)} + \frac{1}{2} \sum_{F \in \mathcal{F}_b} ((\mathcal{M} - \mathcal{D})g, y)_{X(F)}, \quad (52)$$

where $X(\mathcal{F}_{i,b})$ is, for example, $[H^{-\frac{1}{2}}(\mathcal{F}_{i,b})]^d \times [H^{\frac{1}{2}}(\mathcal{F}_{i,b})]^d$ for compressible linear elasticity, $H^{-\frac{1}{2}}(\mathcal{F}_{i,b}) \times H^{\frac{1}{2}}(\mathcal{F}_{i,b})$ for the scalar advection-diffusion-reaction and $L_T^2(\mathcal{F}_{i,b}) \times L_T^2(\mathcal{F}_{i,b})$ for the Maxwell equations in stationary regime, with $L_T^2(\mathcal{F}_{i,b})$ being the space of fields in $H(\text{curl}, \mathcal{T}_h)$ whose tangential component belongs to $[L^2(\mathcal{F}_{i,b})]^3$.

The problem (51) above is well-posed and consistent [12, Lemma 2.4] with the previous formulation in (13a). We consider the optimal norms,

$$\|(z, q)\|_{\mathcal{U}}^2 = \sum_{T \in \mathcal{T}_h} \|z\|_{L(T)}^2 + \|q\|_{\tilde{V}}^2, \quad \|y\|_{\mathcal{Y}}^2 = \sum_{T \in \mathcal{T}_h} \|\tilde{A}y\|_{L(T)}^2 + \|[\![y]\!]\|_{\partial\Omega_h}^2, \quad \text{with } \|[\![y]\!]\|_{\partial\Omega_h} = \sup_{q \in \tilde{V}} \frac{a(q, y)}{\|q\|_{\tilde{V}}}, \quad (53)$$

or formally, considering (52)

$$\|(z, q)\|_{\mathcal{U}}^2 = \sum_{T \in \mathcal{T}_h} \|z\|_{L(T)}^2 + \sum_{F \in \mathcal{F}_{i,b}} \|q\|_{X(F)}^2, \quad \|y\|_{\mathcal{Y}}^2 = \sum_{T \in \mathcal{T}_h} \|\tilde{A}y\|_{L(T)}^2 + \sum_{F \in \mathcal{F}_i} \|\mathcal{D}[\![y]\!]\|_{X(F)}^2 + \sum_{F \in \mathcal{F}_b} \|(\mathcal{M} - \mathcal{D})y\|_{X(F)}^2. \quad (54)$$

With these optimal norms for the trial and test spaces we have the following result [13, Theorem 2.6].

Theorem 4 (Optimally stable formulation). *The bilinear form $b : (L \times \tilde{V}, \|\bullet\|_{\mathcal{U}}) \rightarrow (V(\mathcal{T}_h), \|\bullet\|_{\mathcal{Y}})$ defined as*

$$b(u, y) = \sum_{T \in \mathcal{T}_h} (z, \tilde{A}y)_{L^2(T)} + \sum_{T \in \mathcal{T}_h} \langle Dq, y \rangle_{\tilde{V}(T), V(T)} + \frac{1}{2} \langle (M - D)q, y \rangle_{\tilde{V}, V} \quad (55)$$

with $u = (z, q)$, is an isometry between $L \times \tilde{V}$ and $V'(\mathcal{T}_h)$: we have that $\gamma = \beta = \beta^* = 1$, where

$$\gamma := \sup_{u \in \mathcal{U}} \sup_{y \in \mathcal{Y}} \frac{b(u, y)}{\|u\|_{\mathcal{U}} \|y\|_{\mathcal{Y}}}, \quad \beta := \inf_{u \in \mathcal{U}} \sup_{y \in \mathcal{Y}} \frac{b(u, y)}{\|u\|_{\mathcal{U}} \|y\|_{\mathcal{Y}}}, \quad \beta^* := \inf_{y \in \mathcal{Y}} \sup_{u \in \mathcal{U}} \frac{b(u, y)}{\|u\|_{\mathcal{U}} \|y\|_{\mathcal{Y}}} \quad (56)$$

with $\mathcal{U} = (L \times \tilde{V}, \|\bullet\|_{\mathcal{U}})$.

This property is inherited at the discrete level as long as fixed $U_h^{N_h} \subset \mathcal{U}$ a discretization of the trial space with $\dim Z_h = N_h$, the discrete test space $Y_h^{N_h} \subset V(\mathcal{T}_h)$ is the set of supremizers

$$Y_h^{N_h} = \text{span} \left\{ y_{u_h} \in V(\mathcal{T}_h) \mid y = \underset{y \in V(\mathcal{T}_h)}{\text{argmax}} \frac{b(u_h, y)}{\|y\|_{\mathcal{Y}}}, \quad u_h \in Z_h^{N_h} \right\} \quad (57)$$

and $\dim U_h = \dim Y_h$, see [13, Lemma 2.8]. In particular, for every $u_h = (z_h, q_h) \in U_h^{N_h}$, we have the optimal *a posteriori* error estimate

$$\|u - u_h\|_{\mathcal{U}} = \sup_{y \in V(\mathcal{T}_h)} \frac{b(u - u_h, y)}{\|y\|_{\mathcal{Y}}} = \|r_h(u_h)\|_{(V(\mathcal{T}_h))'}, \quad \langle r_h(u_h), v \rangle_{(V(\mathcal{T}_h))', V(\mathcal{T}_h)} = \langle f, v \rangle_{(V(\mathcal{T}_h))', V(\mathcal{T}_h)} - b(u_h, v). \quad (58)$$

The same reasoning can be iterated another time to perform model order reduction with the choice $V_n = \{\psi_j^{\text{RB}}\}_{j=1}^r \subset U_h^{N_h} \subset \mathcal{U}$, and

$$Y^{\text{RB}} = \text{span} \left\{ y_{u_{\text{RB}}} \in V(\mathcal{T}_h) \mid y = \underset{y \in V(\mathcal{T}_h)}{\text{argmax}} \frac{b(u_{\text{RB}}, y)}{\|y\|_{\mathcal{Y}}}, \quad u_{\text{RB}} \in V_n \right\}, \quad (59)$$

such that for $u_{\text{RB}} \in V_n$,

$$\|u - u_{\text{RB}}\|_{\mathcal{U}} = \|r_{\text{RB}}(u_{\text{RB}})\|_{(V(\mathcal{T}_h))'}, \quad \langle r_{\text{RB}}(u_{\text{RB}}), v \rangle_{(V(\mathcal{T}_h))', V(\mathcal{T}_h)} = \langle f, v \rangle_{(V(\mathcal{T}_h))', V(\mathcal{T}_h)} - b(u_{\text{RB}}, v). \quad (60)$$

The main difficulty is the evaluation of the trial spaces $Y_h^{N_h}$ and Y^{RB} since the bilinear form b may depend on the parameters ρ . If the parameters affect only the source terms, the boundary conditions or the initial conditions for time-dependent FS, this problem is avoided. The evaluation of $Y_h^{N_h}$ can be performed locally for each element $T \in \mathcal{T}_h$, differently from Y^{RB} . An example of the evaluation of the basis of $Y_h^{N_h}$ is presented in [13, Equations 24, 25] for linear scalar hyperbolic equations that can be interpreted as FS.

5 Domain decomposable Discontinuous Galerkin ROMs

Extreme-scale parametric models are unfeasible to reduce with standard approaches due to the high computational costs of the offline stage. Parametric multi-physics simulations, such as fluid-structure interaction problems, are reduced inefficiently with a global reduced basis, depending on the complexity of the interactions between the physical models considered and the parametric dependency. In some cases, only a part of a decomposable system is reducible with a ROM, thus a possible solution is to implement a ROM-FOM coupling through an interface. In presence of moving shocks [7] affected by the parametrization, one may want to isolate these difficult features to approximate and apply different dimension reduction methodologies depending on the subdomain. These are the main reasons to develop domain decomposable or partitioned ROMs (DD-ROMs).

Some approaches from the literature are the reduced basis element methods [61, 62], the static condensation reduced basis element method [48, 27], non-intrusive methods based on local regressions in each subdomain [85, 86], overlapping Schwarz methods [23, 50], optimization-based MOR approaches [70, 71] and hyper-reduced ROMs [60]. In this last case, local approximations are useful because the local reduced dimensions are smaller and therefore more accurate local regressions can be designed to perform non-intrusive surrogate modelling. Little has been developed for the DG method, even though its formulation imposes naturally flux and solution interface penalties at the internal boundaries of the subdomains, in perspective of performing model order reduction. In our case, the linear systems associated to the parametric models are algebraically partitioned in disjoint subdomains coupled with the standard penalties from the weak DG formulations, without the need to devise additional operators to perform the coupling as long as the interfaces' cuts fall on the cell boundaries.

Another less explored feature of DD-ROMs is the possibility to repartition the computational domain, while keeping the data structures relative to each subdomain local in memory, with the aim of obtaining more efficient or accurate ROMs. In fact, one additional reason to subdivide the computational domain is to partition the solution manifold into local solution manifolds that have a faster decay of the Kolmogorov n-width. The repartition of the computational domain can be performed with *ad hoc* domain decomposition strategies. To our knowledge, the only case found in the literature is introduced in [86], where the degrees of freedom are split in each subdomain minimising the communication and activity between them and balancing the computational load across them. Relevant is the choice of weights to assign to each degree of freedom: uniform weights, nodal values of Reynolds stresses for the turbulent Navier-Stokes equations or the largest singular value of the discarded local POD modes. In particular, the last option results in a balance of energy in L^2 norm retained in each subdomain. We explore a different approach.

It must be remarked that in any case, fixed the value of the reconstruction error of the training dataset in Frobenious norm $\|\cdot\|_F$, there must be at least a local reduced basis dimension greater or equal to the global reduced basis space dimension. In fact, for the Eckhart-Young theorem, if $X \in \mathbb{R}^{d \times n}$ is the snapshots matrix ordered by columns, we have that the projection into the first k modes $\{v_k\}_{i=1}^k$ achieves the best approximation error in the Frobenious norm in the space of matrices of rank k :

$$P_k = \underset{P \in \mathbb{R}^{m \times n} \text{ s.t. } r(P)=k}{\text{argmin}} \|X - PX\|_F, \quad P_k = \sum_{i=1}^k v_i \otimes v_i, \quad (61)$$

where $r(\cdot)$ is the matrix rank. So, in general, it is not possible to achieve a better training approximation error in the Frobenious norm $\|\cdot\|_F$ employing a number of local reduced basis smaller than what would be needed to achieve the same accuracy with a global reduced basis. So, differently from [86], instead of balancing the local reduced basis dimension among subdomains, we repartition the computational domain in regions whose restricted solution manifold is easily approximable by linear subspaces and regions for which more modes are needed. Anyway, for truly decomposable systems we expect that the reconstruction error on the test set is lower when considering local reduced basis instead of global ones, as will be shown for the Maxwell equation in stationary regime test case with discontinuous piecewise constant parameters, see Figure 7.

5.1 Implementation of Domain Decomposable ROMs

Let us assume that the full-order model is implemented in parallel with distributed memory parallelism with $K > 1$ cores, i.e., each i -th core owns locally the data structures relevant only to its assigned subdomain Ω_i of the whole computational domain $\cup_{i=1}^K \Omega_i = \Omega_h \subset \mathbb{R}^d$, for $i = 1, \dots, K$. We will employ the **deal.II** library [6] to discretize the FS with the DG method, assemble the associated linear systems and solve them in parallel [9]. In particular, we employ **p4est** [14] to decompose the computational domain, **PETSc** [8] to assemble the linear system and solve it at the full-order level and **petsc4py** [22] to assemble and solve the reduced order system. At the offline and online stages the computations are performed in a distributed memory setting in which each core assembles its own affine decomposition, so that the evaluation of the reduced basis and of the projected local operators is always performed in parallel.

The weak formulation (35) is easily decomposable thanks to the additive properties of the integrals. We recall the definition of the weak formulation $\forall y_h \in V_h, z_h \in V^*$

$$\begin{aligned} a_h^{cf}(z_h, y_h) + s_h(z_h, y_h) &= \sum_{T \in \mathcal{T}_h} (z_h, \tilde{A}y_h)_{L^2(T)} + \frac{1}{2} \sum_{F \in \mathcal{F}_h^b} ((\mathcal{M} + \mathcal{D})z_h, y_h)_{L^2(F)} + \sum_{F \in \mathcal{F}_h^i} (\mathcal{D}_F\{z_h\}, [\![y_h]\!])_{L^2(F)} + \\ &\quad \sum_{F \in \mathcal{F}_h^b} (S_F^b z_h, y_h)_{L^2(F)} + \sum_{F \in \mathcal{F}_h^i} (S_h^i [\![z_h]\!], [\![y_h]\!])_{L^2(F)}, \end{aligned} \quad (62)$$

and we decompose it into the K subdomains as

$$\begin{aligned} a_h^{cf}(z_h, y_h) + s_h(z_h, y_h) &= \sum_{i=1}^K \left(\sum_{T \in \mathcal{T}_{h,i}} (z_h, \tilde{A}y_h)_{L^2(T)} + \frac{1}{2} \sum_{F \in \mathcal{F}_{h,i}^b} ((\mathcal{M} + \mathcal{D})z_h, y_h)_{L^2(F)} + \sum_{F \in \mathcal{F}_{h,i}^i} (\mathcal{D}_F\{z_h\}, [\![y_h]\!])_{L^2(F)} + \right. \\ &\quad \left. \sum_{F \in \mathcal{F}_{h,i}^b} (S_F^b z_h, y_h)_{L^2(F)} + \sum_{F \in \mathcal{F}_{h,i}^i} (S_h^i [\![z_h]\!], [\![y_h]\!])_{L^2(F)} \right) + \sum_{i=1}^K \left(\sum_{j=i}^K \left(\sum_{F \in \mathcal{F}_{h,i,j}^i} (\mathcal{D}_F\{z_h\}, [\![y_h]\!])_{L^2(F)} + \right. \right. \\ &\quad \left. \left. \sum_{F \in \mathcal{F}_{h,i,j}^i} (S_h^i [\![z_h]\!], [\![y_h]\!])_{L^2(F)} \right) \right), \end{aligned} \quad (63)$$

$$l_h(y_h) = \sum_{T \in \mathcal{T}_h} (f, y_h)_{L^2(T)} + \frac{1}{2} \sum_{F \in \mathcal{F}_h^b} ((M - D)g, y_h)_{L^2(F)} = \sum_{i=1}^K \left(\sum_{T \in \mathcal{T}_{h,i}} (f, y_h)_{L^2(T)} + \frac{1}{2} \sum_{F \in \mathcal{F}_{h,i}^b} ((M - D)g, y_h)_{L^2(F)} \right), \quad (64)$$

where we have defined the internal subsets $\mathcal{T}_{h,i} = \mathcal{T}_h \cap \Omega_i$, $\mathcal{F}_{h,i}^i = \mathcal{F}_h^i \cap \mathring{\Omega}_i$ and $\mathcal{F}_{h,i}^b = \mathcal{F}_h^b \cap \mathring{\Omega}_i$, $\forall i = 1, \dots, K$ and the interfaces subsets $\mathcal{F}_{h,i,j}^i = \mathcal{F}_h^i \cap \overline{\Omega}_i \cap \overline{\Omega}_j$ and $\mathcal{F}_{h,i,j}^b = \mathcal{F}_h^b \cap \overline{\Omega}_i \cap \overline{\Omega}_j$, $\forall i = 1, \dots, K$. We remark that the computational domain is always decomposed such that the cuts of the subdomains $\{\partial\Omega_i\}_{i=1}^K$ fall on the interfaces of the triangulation $\mathcal{F}_h^i \cup \mathcal{F}_h^b$.

We define the bilinear and linear operators in V_h^* ,

$$\begin{aligned} \mathcal{A}_{ii} &= \sum_{T \in \mathcal{T}_{h,i}} (\bullet, \tilde{A}\bullet)_{L^2(T)} + \frac{1}{2} \sum_{F \in \mathcal{F}_{h,i}^b} ((\mathcal{M} + \mathcal{D})\bullet, \bullet)_{L^2(F)} + \sum_{F \in \mathcal{F}_{h,i}^i} (\mathcal{D}_F\{\bullet\}, [\![\bullet]\!])_{L^2(F)} + \\ &\quad \sum_{F \in \mathcal{F}_{h,i}^b} (S_F^b \bullet, \bullet)_{L^2(F)} + \sum_{F \in \mathcal{F}_{h,i}^i} (S_h^i [\![\bullet]\!], [\![\bullet]\!])_{L^2(F)}, \quad \forall i = 1, \dots, K, \end{aligned} \quad (65)$$

$$\mathcal{A}_{ij} = \mathcal{A}_{ji} = \sum_{F \in \mathcal{F}_{h,i,j}^i} (\mathcal{D}_F\{\bullet\}, [\![\bullet]\!])_{L^2(F)} + \sum_{F \in \mathcal{F}_{h,i,j}^i} (S_h^i [\![\bullet]\!], [\![\bullet]\!])_{L^2(F)}, \quad \forall j, i = 1, \dots, K, i \neq j, \quad (66)$$

$$\mathcal{F}_i = \sum_{T \in \mathcal{T}_{h,i}} (f, \bullet)_{L^2(T)} + \frac{1}{2} \sum_{F \in \mathcal{F}_{h,i}^b} ((M - D)g, \bullet)_{L^2(F)}, \quad \forall i = 1, \dots, K, \quad (67)$$

and their matrix representation in the discontinuous Galerkin basis of V_h ,

$$(\mathcal{A}_{ii})|_{V_h} = A_{ii}, \quad \mathcal{F}_i|_{V_h} = F_i, \quad \forall i = 1, \dots, K, \quad (\mathcal{A}_{ij})|_{V_h} = (\mathcal{A}_{ji})|_{V_h} = A_{ij} = A_{ji}, \quad \forall j, i = 1, \dots, K, \quad i \neq j, \quad (68)$$

and in the local reduced basis $V_i = \{\psi_{j,i}^{\text{RB}}\}_{j=1}^r \subset V_h(\Omega_i)$, $i = 1, \dots, K$,

$$(\mathcal{A}_{ii})|_{V_{RB}} = B_{ii}, \quad \mathcal{F}_i|_{V_{RB}} = L_i, \quad \forall i = 1, \dots, K, \quad (\mathcal{A}_{ij})|_{V_{RB}} = (\mathcal{A}_{ji})|_{V_{RB}} = B_{ij} = B_{ji}, \quad \forall j, i = 1, \dots, K, \quad i \neq j. \quad (69)$$

As anticipated, in our test cases the subdomains interface penalties are naturally included inside $\{\mathcal{A}_{ij}\}_{i,j=1,\dots,K}$. In practice, additional penalty terms could be implemented:

$$\mathcal{S}_{ij} = \sum_{F \in \mathcal{F}_{h,i,j}^i} (S[\bullet], [\bullet])_{L^2(F)}, \quad \mathcal{S}_{ij}|_{V_h} = S_{ij} \quad (\mathcal{A}_{ij} + \mathcal{S}_{ij})|_{V_{RB}} = B_{ij}, \quad \forall j, i = 1, \dots, K, \quad i \neq j. \quad (70)$$

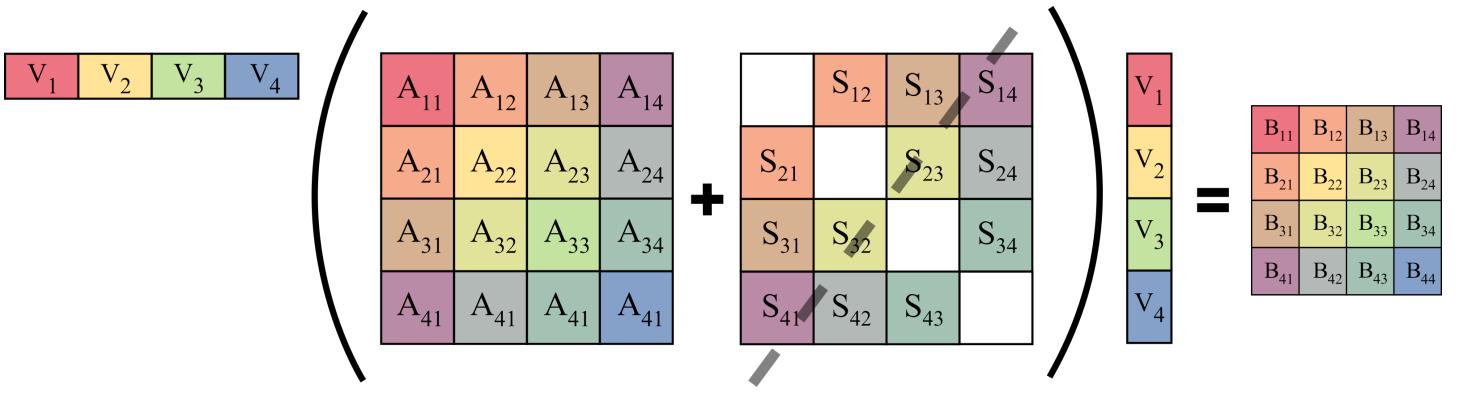


Figure 1: Assembly of the reduced block matrix $\{B_{i,j}\}_{i,j=1}^4$ through the projection onto the local reduced basis $\{V_i\}_{i=1}^4$ of the full-order partitioned matrix $A = \{A_{i,j}\}_{i,j=1}^4$ when considering 4 subdomains. The natural DG penalty terms are included in the matrix A without the need for additional penalty terms $\{S_{i,j}\}_{i \neq j, , i,j=1}^4$ to impose stability at the reduced level.

A matrix representation of the projection of the full-order block matrix $(A_{ij})_{i,j=1}^K \in \mathbb{R}^{d \times d}$ into the reduced order block matrix $(B_{ij})_{i,j=1}^K \in \mathbb{R}^{Kr \times Kr}$ is shown in Figure 1 for $K = 4$. We remark that, differently from continuous Galerkin formulations, the DG penalization on jumps across the interfaces is already enough to couple the subdomains and there is no need of further stabilization, as shown in Figure 1. Nonetheless, additional interface penalties terms can be easily introduced, taking also into account DG numerical fluxes. The reduced dimension is the number of subdomains K times the local reduced basis dimensions $\{r_i\}_{i=1}^K$, here supposed equal $r = r_i$, $i = 1, \dots, K$, but in general can be different.

5.2 Repartitioning strategy

A great number of subdomains can pollute the efficiency of the developed DD-ROMs at the online stage since the reduced dimension would be $\sum_{i=1}^K r_i$ that scales linearly with the number of cores if the local reduced dimensions r_i are equal. In order to keep the computational savings in the assembly of the affine decomposition at the offline stage, we may want to preserve the distributed property of our ROM. One possible solution is to fix a reduced number of subdomains $k \ll K$ such that $\sum_{i=1}^k r_i$ is small enough to achieve a significant speedup with respect to the FOM. The additional cost with respect to a monodomain ROM is associated to the evaluation of the k local reduced basis with SVD and the assembly of the affine decomposition operators. The new k reduced subdomains do not need to be agglomerations of the FOM subdomains, hence, different strategies to assemble the new k reduced subdomains can be investigated.

The number of subdomains K was kept the same as the FOM since it is necessary to collect the snapshots efficiently at the full-order level through **p4est**. However, if we decide to repartition our computational domain, we can develop decomposition strategies that reduce $\sum_{i=1}^K r_i$. Ideally, having in mind the Eckhart-Young theorem, a possible strategy is to lump together all the dofs of the cells that have a fast decaying Kolmogorov n-width, and focus on the remaining ones. We test this procedure in the practical case $k = 2$, $K = 4$ to perform numerical experiments in section 5.3.

To solve the classification problem of partitioning the elements of the mesh into k subdomains, we describe here two scalar indicators that will be used as metrics. For $k = 2$ subdomains, it will be sufficient to choose the percentage of cells P_l corresponding to the lowest values of the chosen scalar indicator. Other strategies for $k > 2$ may also involve clustering

algorithms and techniques to impose connectedness of the clusters, as done for local dimension reduction in parameter spaces in [75]. A first crude and cheap indicator to repartition the computational domain is the cellwise variance of the training snapshots, as it measures how well, in mean squared error, the training snapshots are approximated by their mean, $\forall T \in \mathcal{T}_h$.

Definition 3 (Cellwise variance indicator). *We define the cellwise variace indicator $I_{var} : \mathcal{T}_h \rightarrow \mathbb{R}^+$,*

$$I_{var}(T) = \int_T \|Var(\{\mathbf{z}(\boldsymbol{\rho}_i)\}_{i=1}^n)\|_{L^2(\mathbb{R}^m)} d\mathbf{x}, \quad (Var(\{\mathbf{z}(\boldsymbol{\rho}_i)\}_{i=1}^n))_l = \frac{1}{n} \sum_{i=1}^n \left| \mathbf{z}_l(\boldsymbol{\rho}_i) - \frac{1}{n} \sum_{j=1}^n \mathbf{z}_l(\boldsymbol{\rho}_j) \right|^2, \quad l = 1, \dots, m, \quad (71)$$

where $n > 0$ is the number of training DG solutions $\{\mathbf{z}(\boldsymbol{\rho}_i)\}_{i=1}^n$ with $\mathbf{z}(\boldsymbol{\rho}_i) : \Omega \subset \mathbb{R}^d \rightarrow \mathbb{R}^m$, $\forall i \in \{1, \dots, n\}$.

Note that the indicator is a scalar function on the set of elements of the triangulation \mathcal{T}_h . This is possible thanks to the assumption that boundaries of the subdomains belong to the interfaces of the elements of \mathcal{T}_h . When this hypothesis is not fulfilled, we would need to evaluate additional operators to impose penalties at the algebraical interfaces between subdomains that are not included in the set $\mathcal{F}_h^i \cup \mathcal{F}_h^b$, not to degrade the accuracy.

The cellwise variance indicator is effective for all the test cases for which there is a relatively large region that is not sensitive to the parametric instances, as in our advection diffusion reaction test case in Section 5.3.3. Common examples are all the CFD numerical simulations that have a far field with fixed boundary conditions. However, the variance indicator may be blind to regions in which the snapshots can be spanned by a one or higher dimensional linear subspace and are not well approximated by a constant field, as in the compressible linear elasticity test case in Section 5.3.2.

In these cases, a valid choice is represented by a cellwise Grassmannian dimension indicator. We denote with D_T the number of degrees of freedom associated to each element T , assumed constant in our test cases.

Definition 4 (Cellwise Grassmannian dimension indicator). *Fixed $1 \leq r_T \in \mathbb{N}$, and $1 \leq n_{neig} \in \mathbb{N}$, we define the cellwise Grassmannian dimension indicator $I_G : \mathcal{T}_h \rightarrow \mathbb{R}^+$,*

$$I_G(T) = \|X_T - U_T U_T^T X_T\|_F, \quad (72)$$

where $X_T \in \mathbb{R}^{n_{neig} D_T \times n}$ is the snapshots matrix restricted to the cell T and its n_{neig} nearest neighbours, and $U_T \in \mathbb{R}^{n_{neig} D_T \times r_T}$ are the modes of the truncated SVD of X_T with dimension r_T .

The cellwise Grassmannian dimension indicator I_G is a measure of how well the training snapshots restricted to a neighbour of each cell are approximated by a r_T dimensional linear subspace. Employing this indicator, we recover an effective repartitioning of the computational subdomain of the compressible linear elasticity test case, see Section 5.3.2. The Grassmannian indicator has two hyper-parameters that we fix for each test case in section 5.3: the number of nearest-neighbour cells is $n_{neig} = 3$ and the number of reduced local dimension used to evaluate the L^2 reconstruction error is $r_T = 1$. The number of nearest-neighbour is chosen to deal with critical cases at the boundaries and the closest neighbouring cells are chosen based on the distance of barycenters. The reduced local dimension $r_T = 1$ is chosen very small as the computations must be done on very few cells.

We remark that both indicators do not guarantee that the obtained subdomains belong to the same connected components and, though this might be a problem in terms of connectivity and computational costs for the FOM, at the reduced level this does not affect the online computational costs. Nevertheless, in the tests we perform, the obtained subdomains are connected.

Now, the assembly of the affine decomposition proceeds as explained in Section 5.1 with the difference that at least one local reduced basis and reduced operator is split between at least 2 subdomains/cores. A schematic block matrix representation of the procedure is shown in Figure 2.

5.3 Numerical experiments

In this section, we test the presented methodology for different linear parametric partial differential equations: the Maxwell equations in stationary regime in section 5.3.1 (**MS**), the compressible linear elasticity equations in section 5.3.2 (**CLE**) and the advection diffusion reaction equations in section 5.3.3 (**ADR**). We study two different parametrizations for the test cases **MS** and **CLE**: one with parameters that affect the whole domain **MS1** and **CLE1**, and one with parameters that affect independently different subdomains **MS2** and **CLE2**. We show a case in which DD-ROMs work effectively **MS2** and a case **CL2** in which the performance is analogous to single domain ROMs, even if the parameters have a local influence.

We test the effectiveness of the *a posteriori* error estimates introduced in section 4.1, the accuracy of DD-ROMs for $K = 4$ and the results of repartitioning strategies with $k = 2$ subdomains. When performing a repartition of the computational domain Ω in subdomains $\{\Omega_i\}_{i=1}^k$ with reduced dimensions $\{r_{\Omega_i}\}_{i=1}^k$, we call the subdomains with lower values of the variance indicator I_{var} , see definition 3, *low variance* regions and with lower values of the Grassmannian indicator I_G , see definition 4,

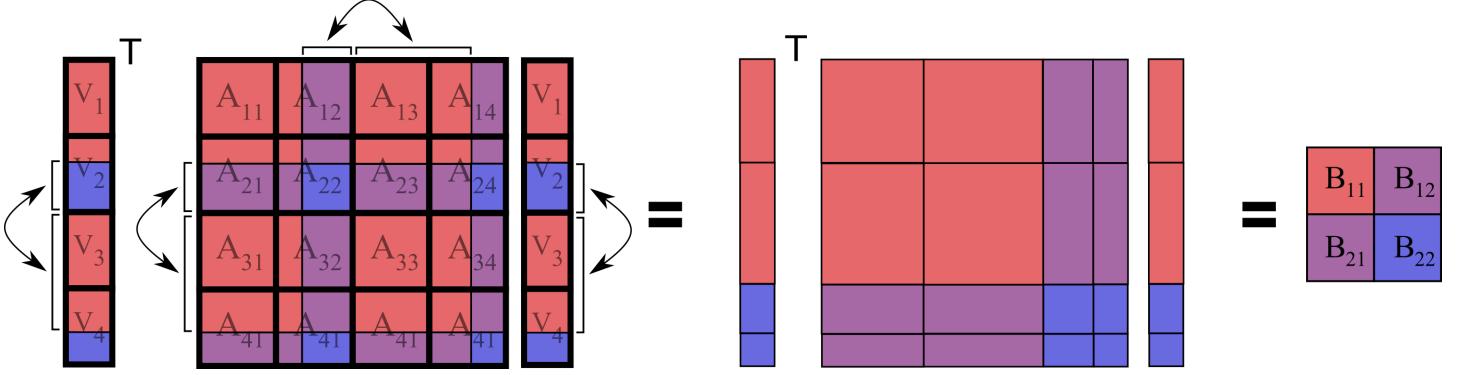


Figure 2: Repartitioning of the reduced block matrix shown in Figure 1 from $K = 4$ subdomains to $k = 2$ repartitioned subdomains. The projection from the full-order matrix $A = \{A_{i,j}\}_{i,j=1}^4$ to the reduced matrix $B = \{B_{i,j}\}_{i,j=1}^2$ is sketched. It is performed locally in a distributed memory setting, the re-ordering shown by the arrows is reported only to visually see the which block structure of the full-order matrix A would correspond to the blocks of the reduced matrix B .

low Grassmannian reconstruction error. The complementary subdomains are the *high variance* and *high Grassmannian reconstruction error* regions, respectively. We show a case (**CLE1**) in which the Grassmannian indicator detects a better partition in terms of local reconstruction error with respect to the variance indicator.

We will observe that the relative errors in R -norm and energy norm and the L^2 relative error estimator and L^2 relative energy norm estimator are the most affected by the domain partitions.

The open-source software library employed for the implementation of the full-order Friedrichs' systems discontinuous Galerkin solvers is **deal.II** [6] and we have used piecewise \mathbb{P}^2 basis functions in all simulations. The partition of the computational domain is performed in **deal.II** through the open-source **p4est** package [14]. The distributed affine decomposition data structures are collected in the offline stage and exported in the sparse NumPy format [41]. The reduced order models and the repartition of the computational domains are implemented in Python with MPI-based parallel distributed computing **mpi4py** [21] and **petsc4py** [8] for solving the linear full-order systems through **MUMPS** [1], a sparse direct solver.

5.3.1 Maxwell equations in stationary regime (MS)

We consider the parametric Maxwell equations in the stationary regime in $d = 3$ spatial dimensions, with $m = 6$ equations, on a torus $\Omega \subset \mathbb{R}^3$ with inner radius $r = 0.5$ and outer radius $R = 2$ centered in $\mathbf{0}$ and lying along the (x, z) plane:

$$\begin{pmatrix} \mu \mathbf{H} + \nabla \times \mathbf{E} \\ \sigma \mathbf{E} - \nabla \times \mathbf{H} \end{pmatrix} = \begin{pmatrix} \mathbf{g} \\ \mathbf{f} \end{pmatrix}, \quad \forall \mathbf{x} \in \Omega, \quad (73)$$

the tangential homogeneous boundary conditions $\mathbf{n} \times \mathbf{E} = \mathbf{0}$ are applied with the boundary operator (18). We vary the parameters in the interval $\boldsymbol{\rho} = (\mu, \sigma) \in [0.5, 2] \times [0.5, 3] \subset \mathbb{R}^2$, leading to $\mu_0 = \min(\mu, \sigma)$.

We consider the exact solutions

$$\begin{aligned} \mathbf{H}_{\text{exact}}(\mathbf{x}) &= -\frac{1}{\mu} \left(\frac{2xy}{\sqrt{x^2 + z^2}}, \frac{-4y^2 \sqrt{x^2 + z^2} + \sqrt{x^2 + z^2}(-12(x^2 + z^2) - 15) + 32(x^2 + z^2)}{4(x^2 + z^2)}, \frac{2xy}{\sqrt{x^2 + z^2}} \right), \\ \mathbf{E}_{\text{exact}}(\mathbf{x}) &= \left(\frac{z}{\sqrt{x^2 + z^2}}, 0, -\frac{x}{\sqrt{x^2 + z^2}} \right) \cdot \left(r^2 - y^2 - (R - \sqrt{x^2 + z^2})^2 \right). \end{aligned}$$

We remark that the exact solutions can be approximated with a linear reduced subspace of dimension 1, if we obtained the reduced basis with a partitioned SVD on the fields (\mathbf{H}, \mathbf{E}) separately. We do not choose this approach and perform a monolithic SVD to test the convergence of the approximation with a DD-ROMs with respect to the local reduced dimensions. The source terms are defined consequently as

$$\mathbf{g}(\mathbf{x}) = 0, \quad \mathbf{f}(\mathbf{x}) = \sigma \mathbf{E}_{\text{exact}} - \nabla \times \mathbf{H}_{\text{exact}}. \quad (74)$$

We consider two parametric spaces:

$$\boldsymbol{\rho} = (\mu, \sigma) \in [0.5, 2] \times [0.5, 3] = \mathcal{P}_1 \subset \mathbb{R}^2, \quad (\mathbf{MS1}) \quad (75a)$$

$$\boldsymbol{\rho} = (\mu_1, \sigma_1, \mu_2, \sigma_2) \in [0.5, 2] \times [0.5, 3] \times [0.5, 2] \times [0.5, 3] = \mathcal{P}_2 \subset \mathbb{R}^4, \quad (\mathbf{MS2}) \quad (75b)$$

where in the second case, the parameters μ and σ are now piecewise constant:

$$\mu = \begin{cases} \mu_1, & x < 0, \\ \mu_2, & x \geq 0, \end{cases} \quad \sigma = \begin{cases} \sigma_1, & x < 0, \\ \sigma_2, & x \geq 0, \end{cases} \quad (76)$$

where $\mathbf{x} = (x, y, z) \in \Omega \subset \mathbb{R}^3$. In Figure 3, we show solutions for $\mu = \sigma = 1$ and for discontinuous values of the parameters: $\mu_1 = \sigma_1 = 1$ in $\{x < 0\} \cap \Omega$ and $\mu_2 = \sigma_2 = 2$ in $\{x \geq 0\} \cap \Omega$. The FOM partitioned and DD-ROM repartitioned subdomains are shown in Figure 4. For **MS1**, we choose the variance indicator to repartition the computational subdomain in two subsets: $P_l = 20\%$ of the cells for the *low variance* part and 80% for the *high variance* part. For **MS2**, we split the computational domain in two parts with the Grassmannian indicator and $P_l = 50\%$.

At the end of this subsection a comparison of the effectiveness of DD-ROMs with and without discontinuous parameters will be performed, the associated error plots are reported in Figure 6 and Figure 7. We will see that, for this simple test case **MS2**, there is an appreciable improvement of the accuracy when the computational domain subdivisions match the regions $\{x < 0\} \cap \Omega$ and $\{x \geq 0\} \cap \Omega$ in which μ and σ are constant. Such subdivision is detected by the Grassmannian indicator with $P_l = 50\%$, as shown in Figure 4 on the right. This is the archetypal case in which DD-ROMs are employed successfully, in comparison with **MS1** for which there is no significant improvement with respect to classical global linear reduced basis.

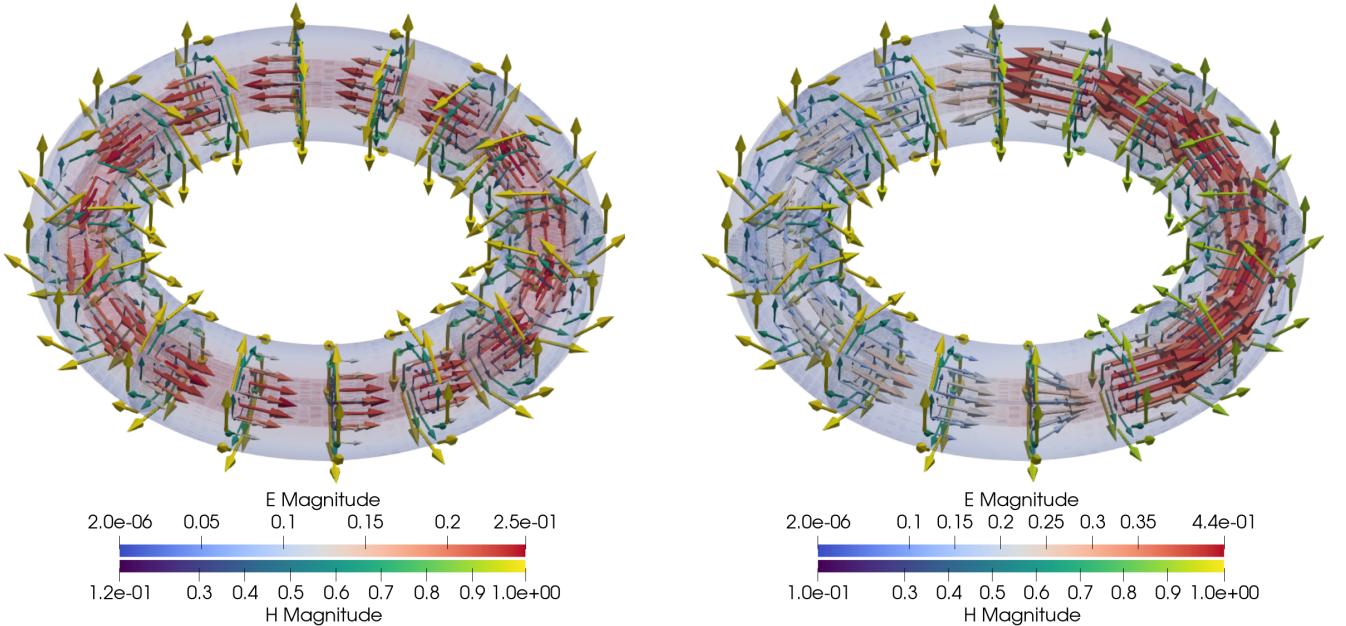


Figure 3: **MS**. Electric and magnetic fields of the Maxwell equations in stationary regime with Dirichlet homogeneous boundary conditions $\mathbf{n} \times \mathbf{E} = \mathbf{0}$. The vectors of the magnetic and electric fields are scaled by 0.5 and 2 of their magnitude respectively. **Left:** **MS1**, $\mu = \sigma = 1$, test case errors shown in Figure 6. **Right:** **MS2**, $\mu = \sigma = 1$ in $\{x < 0\} \cap \Omega$ and $\mu = \sigma = 2$ in $\{x \geq 0\} \cap \Omega$, test case errors shown in Figure 7.



Figure 4: **MS**. **Left:** FOM computational domain partitioned in $K = 4$ subdomains inside `deal.II`. **Center:** **MS1**, DD-ROM repartition of the computational subdomain $k = 2$ with the cellwise variance indicator I_{var} , definition 3: 20% of the cells belong to the *low variance* part, represented in blue inside the torus, and the other 80% belong to the *high variance* part, represented in red. **Right:** **MS2**, DD-ROM repartition with variance indicator $P_l = 50\%$. The computational domain is exactly split at the interfaces that separate the subdomains $\{x < 0\} \cap \Omega$ and $\mu = \sigma = 2$ in which the parameters μ and σ are constant.

In Figure 5, we show how the different thresholds applied to the two indicators can affect the reconstruction error on a reduced space with $r_{\Omega_i} = 3$. All the lines plot the local relative error computed on different subdomains (either one of the k DD-ROM subdomains or on the whole domain). On the x -axis it is shown the percentage of cells that are grouped into the

low variance or low Grassmannian DD-ROM subdomain. We observe that the cellwise variance indicator is a good choice for the purpose of repartitioning the subdomain from $K = 4$ to $k = 2$. Indeed, it is possible to build a low variance subdomain (value of the abscissa 20% in Figure 5) with a low local relative reconstruction error ($5 \cdot 10^{-4}$) with respect to the global one ($8.6 \cdot 10^{-4}$). This means that choosing the threshold $P_l = 20\%$ for the low variance subdomain, we should be able to use less reduced basis functions for that subdomain without affecting too much the global error.

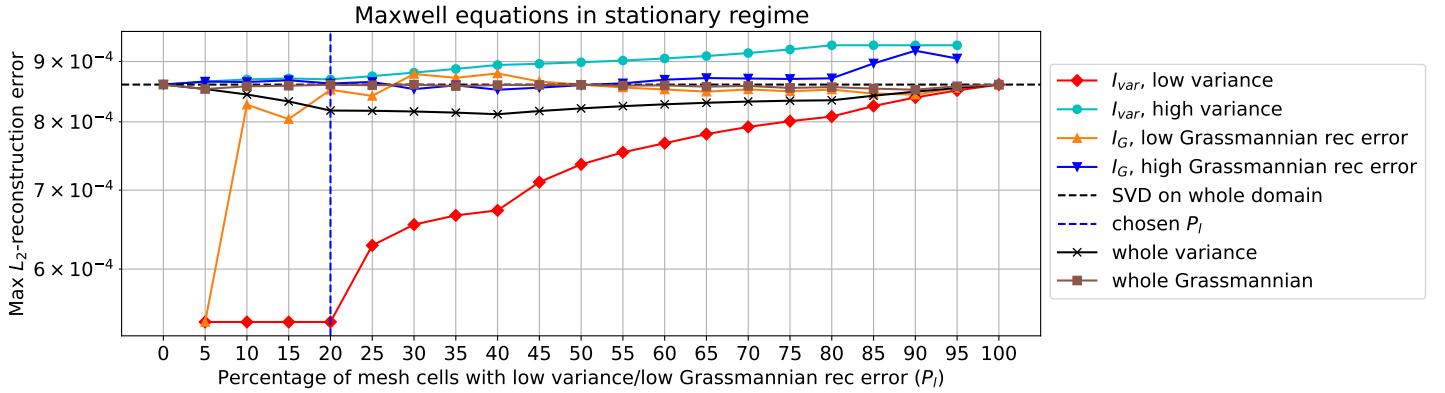


Figure 5: **MS1.** Local relative L^2 -reconstruction errors of the snapshots matrix restricted to the two subdomains of the repartitioning performed with the indicator I_{var} (in red and light-blue), Definition 3, and I_G (in orange and blue), Definition 4. The relative L^2 -reconstruction error attained on the whole domain is shown in black for the indicator I_{var} and in brown for the indicator I_G . The local reduced dimensions used to evaluate the local reconstruction errors is $r_{\Omega_i} = 3$, $i = 1, 2$.

Test case MS1. We evaluate $n_{\text{train}} = 20$ training full-order solutions and $n_{\text{test}} = 80$ test full-order solutions, corresponding to a uniform independent sampling from the parametric domain $\mathcal{P}_1 \subset \mathbb{R}^2$. Figure 6 shows the result relative to the relative L^2 -error and relative errors in energy norm, with associated *a posteriori* estimators. The numberd abscissae $0, 5, 10, \dots, 95$ represents the train parameters $n_{\text{train}} = 20$ while the others $n_{\text{test}} = 80$ parameters are the test set. For these studies, we have fixed the local reduced dimensions to $r_{\Omega_i} = 3$, $i = 1, \dots, K$ for $K = 4$, $r_{\Omega} = 3$ for the whole computational domain and $r_{\Omega_1} = 2$, $r_{\Omega_2} = 3$ for the DD-ROM repartitioned case with $k = 2$. This choice of repartitioning with the 20% of *low variance* cells and local reduced dimension $r_{\Omega_1} = 2$ does not deteriorate significantly the accuracy and the errors almost coincide for all approaches. However, unless the parameters σ, μ assume different discontinuous values in the computational domain Ω , DD-ROMs are not advisable for this test case if the objective is improving the predictions' accuracy.

Test case MS2. Similarly to the previous case, we evaluate $n_{\text{train}} = 20$ training full-order solutions and $n_{\text{test}} = 80$ test full-order solutions, corresponding to a uniform independent sampling from the parametric domain $\mathcal{P}_2 \subset \mathbb{R}^4$. As mentioned above, if we vary the parameters $\rho = (\mu, \sigma)$ discontinuously on the subdomains $\{x \geq 0\} \cap \Omega$ and $x \in \{x < 0\} \cap \Omega$, we obtain the results shown in Figure 7. It can be seen that repartitioning Ω in $k = 2$ DD-ROM subdomains with the local Grassmannian indicator I_G and $P_l = 50\%$ produces effective DD-ROMs compared to the case of a single reduced solution manifold for the whole computational domain and for the DD-ROM with $k = 4$ for which the subdomains do not match $\{x < 0\} \cap \Omega$ and $\{x \geq 0\} \cap \Omega$. In this case, we kept the local dimension of DD-ROM repartitioned case with $k = 2$ equal $r_{\Omega_1} = r_{\Omega_2} = 3$. For this simple test case, there is an appreciable improvement for some test parameters in the accuracy for $k = 2$ instead of $K = 4$ or a classical global linear basis ROM.

In Table 1, we list the computational times and speedups for a simulation with the different methods. For an error convergence analysis with respect to the size of the reduced space, we refer to Appendix C.

Table 1: **MS1.** Average computational times and speedups for ROM and DD-ROM approaches for Maxwell equations. The speedup is computed as the FOM computational time over the ROM one. The FOM runs in parallel with 4 cores, so “FOM time” refers to wallclock time.

FOM		ROM			DD-ROM		
N_h	time	r	time	speedup	r_i	time	speedup
6480	254.851 [ms]	3	51.436 [μ s]	~ 495	[3, 3, 3, 3]	62.680 [μ s]	~ 406

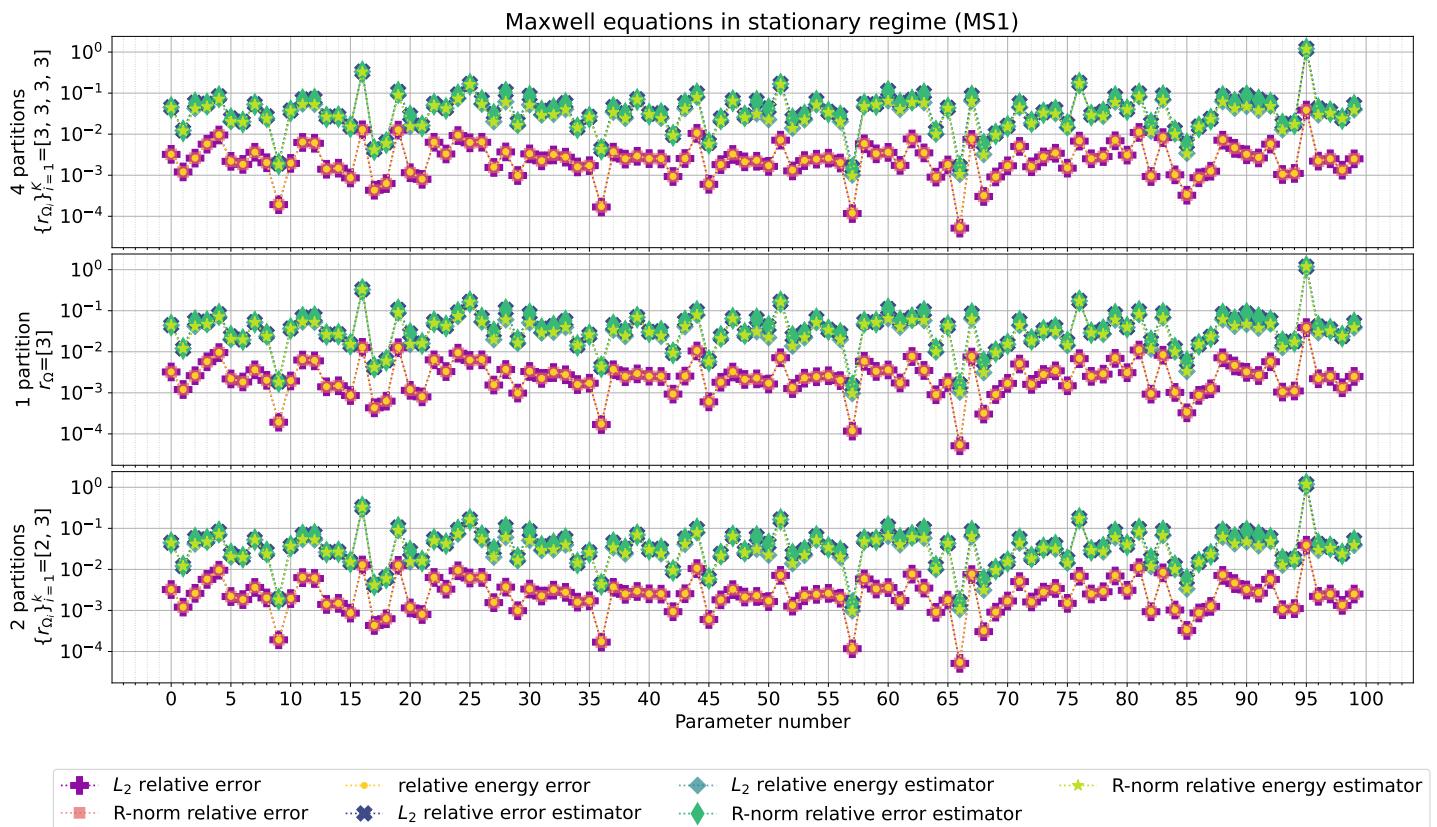


Figure 6: **MS1.** Errors and estimators for Maxwell equations corresponding to the $n_{\text{train}} = 20$ uniformly sampled training snapshots corresponding to the abscissae $0, 5, 10, \dots, 95$, and $n_{\text{test}} = 80$ uniformly sampled test snapshots, corresponding to the other abscissae. The reduced dimensions of the ROMs are $\{r_{\Omega_i}\}_{i=1}^K = [3, 3, 3, 3]$ for $K = 4$ partitions, $r_{\Omega} = 3$ for $k = 1$ partition, and $\{r_{\Omega_i}\}_{i=1}^k = [2, 3]$ for $k = 2$ partitions. For the case $k = 2$ we employed the cellwise variance indicator I_G , Definition 3, with $P_l = 20\%$. It can be seen that even reducing the local dimension from 3 to 2 of one of the $k = 2$ repartitioned subdomains, the accuracy of the predictions does not decrease sensibly.

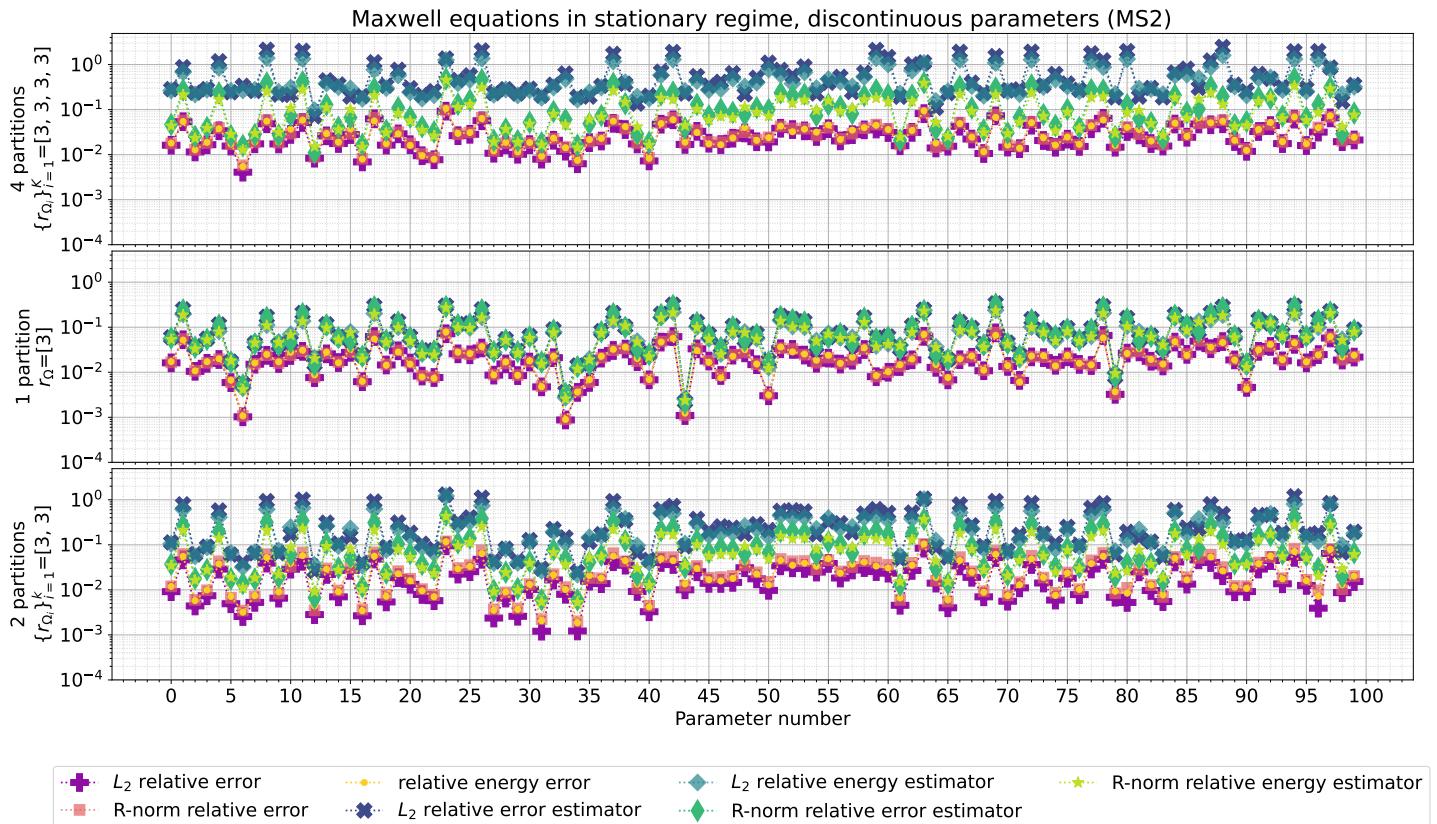


Figure 7: **MS2.** Errors and estimators for Maxwell equations with discontinuous μ and σ corresponding to the $n_{\text{train}} = 20$ uniformly sampled training snapshots corresponding to the abscissae $0, 5, 10, \dots, 95$, and $n_{\text{test}} = 80$ uniformly sampled test snapshots, corresponding to the other abscissae. The reduced dimensions of the ROMs are $\{r_{\Omega_i}\}_{i=1}^K = [3, 3, 3, 3]$ for $K = 4$ partitions, $r_{\Omega} = 3$ for $k = 1$ partition, and $\{r_{\Omega_i}\}_{i=1}^k = [3, 3]$ for $k = 2$ partitions. For the case $k = 2$ we employed the cellwise local Grassmannian dimension indicator I_G , Definition 3, with $P_l = 50\%$. The subdivisions detected exactly match the subdomains $\{x < 0\} \cap \Omega$ and $\{x \geq 0\} \cap \Omega$ in which the parameters are constant. An improvement of the predictions can be appreciated for some test parameters when employing $k = 2$ repartitions.

5.3.2 Compressible linear elasticity (CLE)

Next, we consider the parametric compressible linear elasticity system in $d = 3$ physical dimensions with a cylindrical shell along the z-axis as domain: the inner radius is 1, outer radius 3 and height 10, and the base centered in $\mathbf{0}$. The $m = 12$ equations of the FS are

$$\begin{pmatrix} \boldsymbol{\sigma} - \mu_1(\nabla \cdot \mathbf{u})\mathbb{I}_3 - 2\mu_2 \frac{(\nabla \mathbf{u} + (\nabla \mathbf{u})^t)}{2} \\ -\frac{1}{2}\nabla \cdot (\boldsymbol{\sigma} + \boldsymbol{\sigma}^t) + \mu_3 \mathbf{u} \end{pmatrix} = \begin{pmatrix} \mathbf{0} \\ \mathbf{f} \end{pmatrix}, \quad \forall \mathbf{x} \in \Omega \subset \mathbb{R}^3, \quad (77)$$

where $\boldsymbol{\rho} = (\mu_1, \mu_2, \mu_3) \in [100, 1000]^2 \times [1, 1] = \mathcal{P} \subset \mathbb{R}^3$ and $\mathbf{f} = (0, -1, 0)$. The system can be rewritten as FS as in (20). We define the boundaries

$$\Gamma_D = \partial\Omega \cap \{z = 0\}, \quad \Gamma_N = \partial\Omega \setminus \Gamma_D. \quad (78)$$

Mixed boundary conditions are applied with the boundary operator (24): homogeneous Dirichlet boundary conditions are imposed on Γ_D and homogeneous Neumann boundary conditions on Γ_N .

We consider two parametric spaces:

$$\boldsymbol{\rho} = (\mu_1, \mu_2) \in [100, 1000]^2 = \mathcal{P}_1 \subset \mathbb{R}^2, \quad (\text{CLE1}) \quad (79a)$$

$$\boldsymbol{\rho} = (\mu_1, \mu_2, f_1, f_2) \in [100, 1000]^2 \times [-2, 2]^2 = \mathcal{P}_2 \subset \mathbb{R}^4, \quad (\text{CLE2}) \quad (79b)$$

where in the second case, the source term \mathbf{f} is now piecewise constant:

$$\mathbf{f} = \begin{cases} f_1 \cdot (0, -1, 0), & z < 5, \\ f_2 \cdot (0, -1, 0), & z \geq 5. \end{cases} \quad (80)$$

We show two sample solutions for $\mu_1 = \mu_2 = 1000$ in Figure 8 for **CLE1** and $\mu_1 = \mu_2 = 1000$, $f_1 = 1$ and $f_2 = -1$ for **CLE2**, on the left and on the right, respectively. The partitioned and repartitioned subdomains are shown in Figure 9. For the first case **CLE1** we employ a mesh of 24 cells and 7776 dofs, for the second **CLE2** a mesh of 60 cells and 19440 dofs.

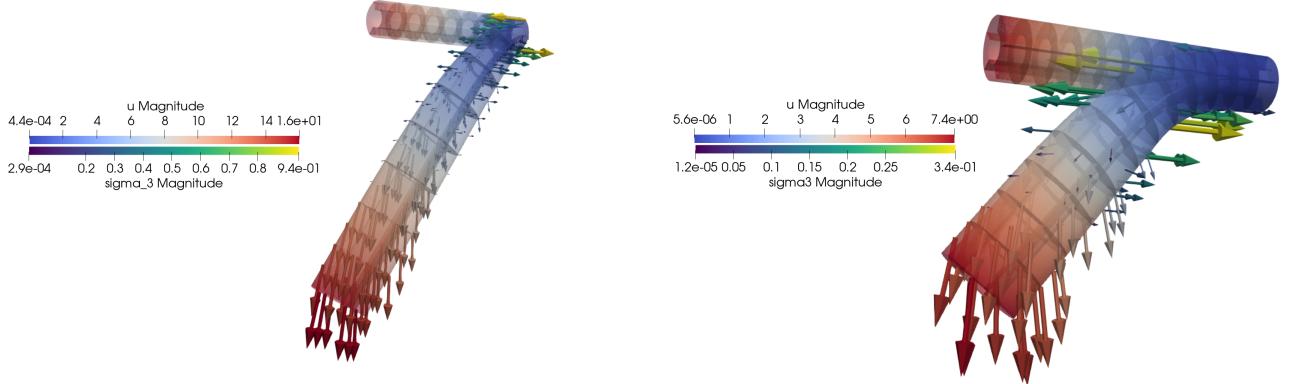


Figure 8: **CLE. Left:** solution of the compressible linear elasticity FS **CLE1** with parameter values $\mu_1 = \mu_2 = 1000$. The cylindrical shell displacement \mathbf{u} , and with a different colorbar also the field $\sigma_{\mathbf{e}_z}$, named sigma_3 , are shown. At the extremity close to $z = 0$ homogeneous Dirichlet boundary conditions are imposed. **Right:** solution of the test case **CLE2** with discontinuous values of the source terms along the computational domain $\{z < 5\} \cap \Omega$ and $\{z \geq 5\} \cap \Omega$: $\mu_1 = \mu_2 = 1000$, $f_1 = 1$ and $f_2 = -1$.

Test case CLE1. This test case presents no region for which the restricted solutions are more or less approximable with a constant field, as would be detected by the variance indicator: as shown in Figure 10, the local relative L^2 -reconstruction error in the region with *low variance*, assigned by I_{var} , deteriorates from the value $2 \cdot 10^{-3}$ of the abscissae 0% and 100% to $1 \cdot 10^{-2}$ of the abscissae 4%. Nonetheless, despite the parametric solutions are not approximable efficiently with a constant field, they are well represented by a one dimensional linear subspace in the region located by the cellwise Grassmannian dimension indicator I_G , for $P_l = 12\%$. The associated *low local Grassmannian dimension* region for $P_l = 12\%$ is shown in Figure 9 in blue.

Also in this test case, the employment of DD-ROMs is not advisable, since there are little gains in the local relative L^2 -reconstruction error for the *low local Grassmannian dimensional* region (values around $3 \cdot 10^{-3}$, in orange for the abscissa $P_l = 12\%$, in Figure 10). The choice of local reduced dimensions $r_{\Omega_1} = 2$ and $r_{\Omega_2} = 3$ does not affect greatly the errors shown in Figure 11. Also in this case, we evaluate $n_{\text{train}} = 20$ training full-order solutions and $n_{\text{test}} = 80$ test full-order solutions,



Figure 9: **CLE.** **Left:** computational subdomains partitioned in $K = 4$ subdomains by `petsc4py` inside `deal.II`. **Center:** test case **CLE1** repartition of the computational subdomain $k = 2$ with the cellwise Grassmannian dimension indicator I_G , Definition 4: 12% of the cells belong to the *low local Grassmannian dimension* part, represented in blue inside the torus, and the other 88% belong to the *high local Grassmannian dimension* part, represented in red. **Right:** test case **CLE2** repartition of the computational subdomain $k = 2$ with the cellwise Grassmannian dimension indicator I_G and $P_l = 50\%$.

corresponding to a uniform independent sampling from the parametric domain $\mathcal{P} \subset \mathbb{R}^3$. Also for these studies, we have fixed the local dimensions to $r_{\Omega_i} = 3$, $i = 1, \dots, K$ for $K = 4$, $r_{\Omega} = 3$ for the whole computational domain and $r_{\Omega_1} = 2$, $r_{\Omega_2} = 3$ for the repartitioned case with $k = 2$.

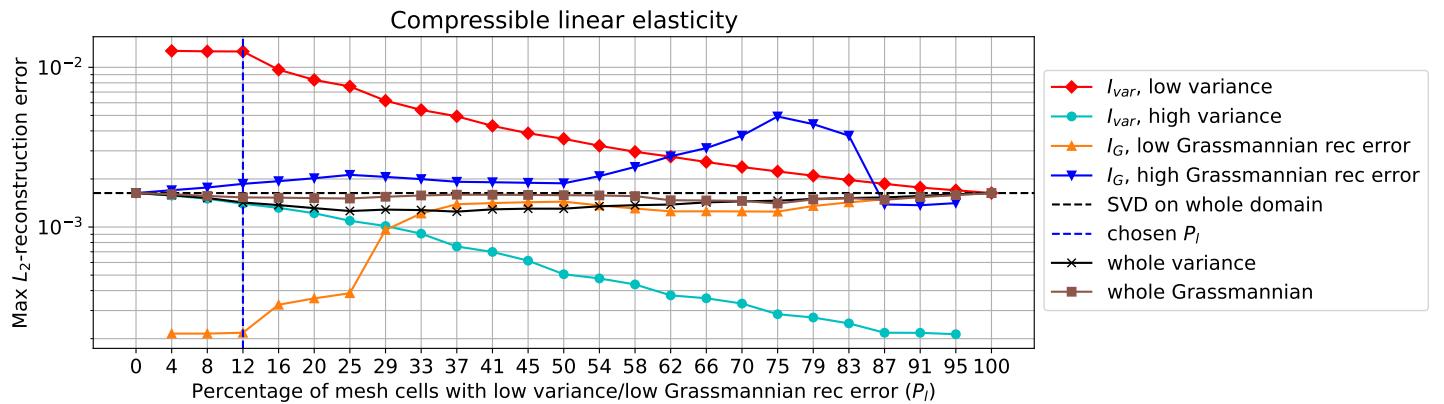


Figure 10: **CLE1.** Local relative L^2 -reconstruction errors of the snapshots matrix for elasticity equations restricted to the two subdomains of the repartitioning performed with the indicator I_{var} (in red and light-blue), Definition 3, and I_G (in orange and blue), Definition 4. The relative L^2 -reconstruction error attained on the whole domain is shown in black for the indicator I_{var} and in brown for the indicator I_G . The local reduced dimensions used to evaluate the local reconstruction errors is $r_{\Omega_i} = 3$, $i = 1, 2$.

Test case CLE2. Similarly to the previous case, we evaluate $n_{\text{train}} = 20$ training full-order solutions and $n_{\text{test}} = 80$ test full-order solutions, corresponding to a uniform independent sampling from the parametric domain $\mathcal{P}_2 \subset \mathbb{R}^4$. This time, if we vary the parameters f_1 and f_2 inside different subdomains $\{z \geq 5\} \cap \Omega$ and $\{z < 5\} \cap \Omega$, we obtain the results shown in Figure 12. It can be seen that repartitioning Ω in $k = 2$ DD-ROM subdomains with the local Grassmannian indicator I_G and $P_l = 50\%$ does not produce more accurate DD-ROMs compared to the case of a single reduced solution manifold for the whole computational domain and for the DD-ROM with $k = 4$. In this case, we kept the local dimension of DD-ROM repartitioned case with $k = 2$ equal $r_{\Omega_1} = r_{\Omega_2} = 3$. For this simple test case, there is not an appreciable improvement for some test parameters in the accuracy for $k = 2$ instead of $K = 4$ or a classical global linear basis ROM. The reason is that even if the parameters f_1 and f_2 affect different subdomains of Ω , the solutions on the whole domain are still well correlated. Differently from the previous test case **MS2** from section 5.3.1, this is a typical case for which DD-ROMs are not effective, even if the parametrization affects independently two regions of the whole domain Ω .

In Table 2, we list the computational times and speedups for a simulation with the different methods. For an error analysis with respect to the size of the reduced space, we refer to Appendix C.

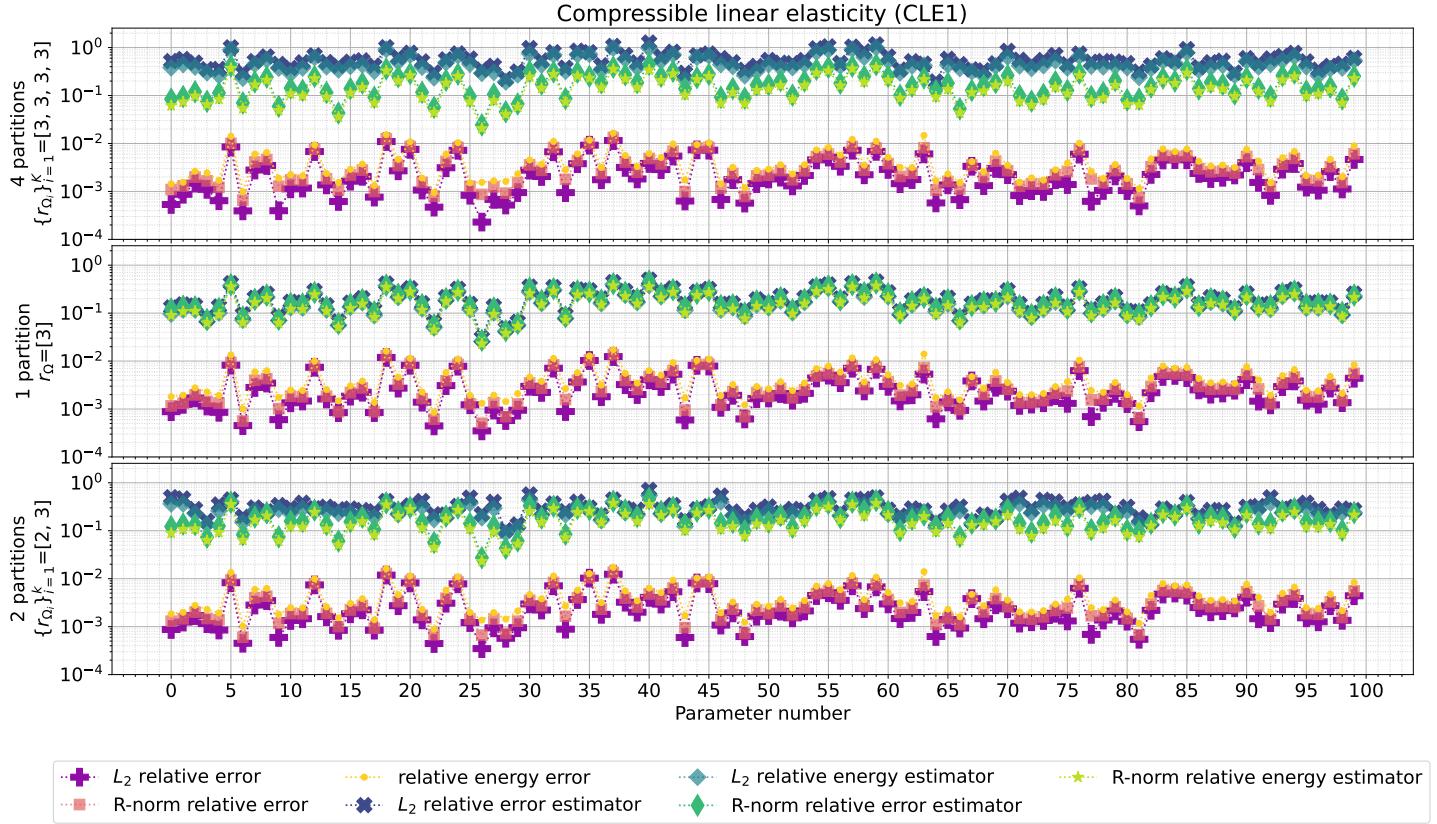


Figure 11: **CLE1.** Errors and estimators for elasticity equations corresponding to the $n_{\text{train}} = 20$ uniformly sampled training snapshots corresponding to the abscissae $0, 5, 10, \dots, 95$, and $n_{\text{test}} = 80$ uniformly sampled test snapshots, corresponding to the other abscissae. The reduced dimensions of the ROMs are $\{r_{\Omega_i}\}_{i=1}^K = [3, 3, 3, 3]$ for $K = 4$ partitions, $r_{\Omega} = 3$ for $k = 1$ partition, and $\{r_{\Omega_i}\}_{i=1}^k = [2, 3]$ for $k = 2$ partitions. For the case $k = 2$ we employed the cellwise local Grassmannian dimension indicator I_G , Definition 4, with $P_l = 12\%$.

Table 2: **CLE.** Average computational times and speedups for ROM and DD-ROM approaches for Maxwell equations. The speedup is computed as the FOM computational time over the ROM one. The FOM runs in parallel with 4 cores, so “FOM time” refers to wallclock time. The first row correspond to test case **CLE1**, the second to test case **CLE2**.

FOM		ROM			DD-ROM		
N_h	time	r	time	speedup	r_i	time	speedup
7776	411.510 [ms]	3	80.444 [μ s]	~ 5115	[3, 3, 3, 3]	85.108 [μ s]	~ 4835
19440	2.080 [s]	3	69.992 [μ s]	~ 29718	[3, 3, 3, 3]	94.258 [μ s]	~ 22067

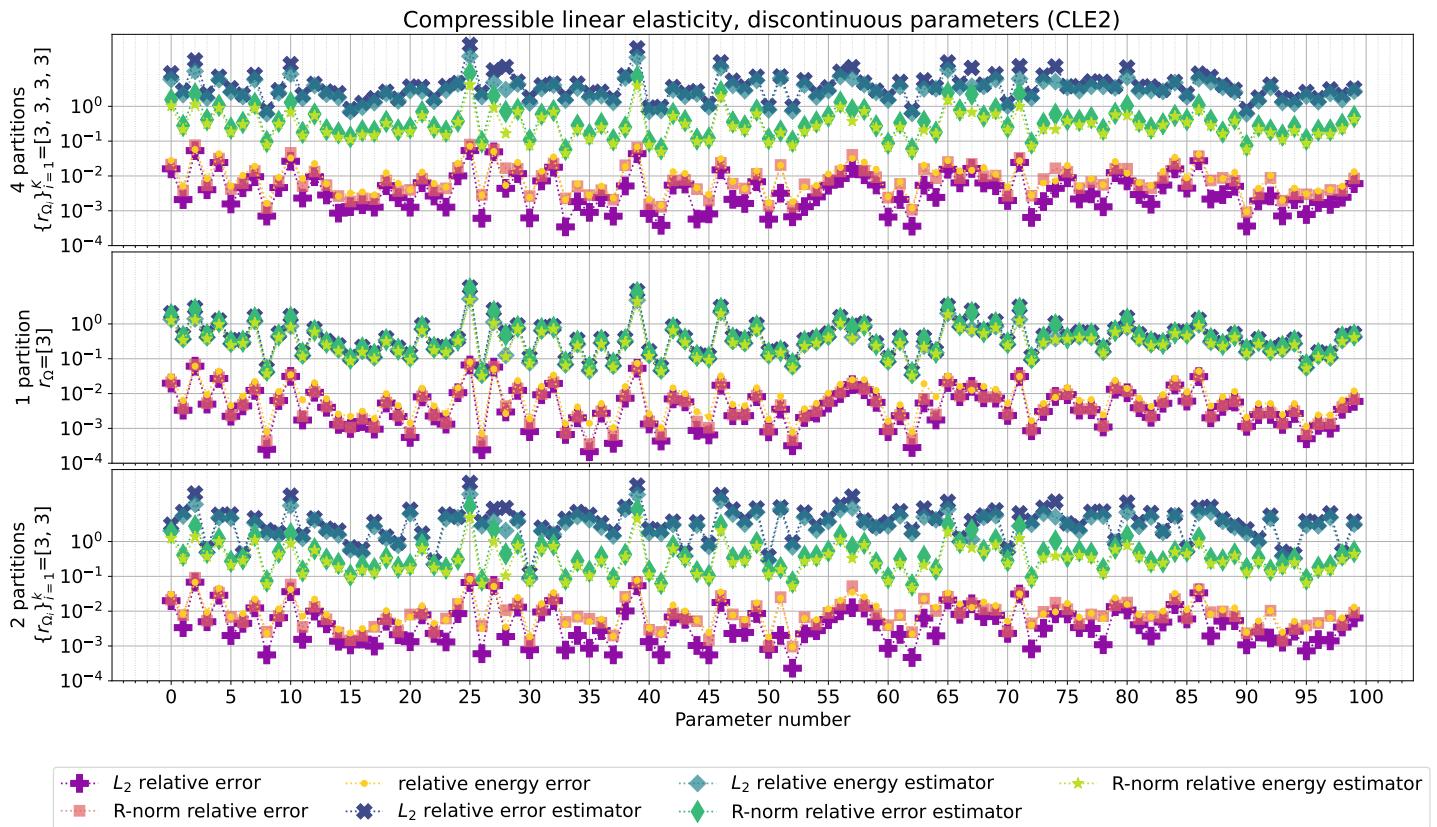


Figure 12: **CLE2.** Errors and estimators corresponding to the $n_{\text{test}}=80$ uniformly sampled test snapshots corresponding to the abscissae $0, 5, 10, \dots, 95$, and $n_{\text{train}}=20$ uniformly sampled training snapshots, corresponding to the other abscissae. The reduced dimensions of the ROMs are $\{r_{\Omega_i}\}_{i=1}^K = [3, 3, 3]$ for $K = 4$ partitions, $r_{\Omega} = 3$ for $k = 1$ partition, and $\{r_{\Omega_i}\}_{i=1}^k = [3, 3]$ for $k = 2$ partitions. For the case $k = 2$ we employed the cellwise variance dimension indicator I_{var} , Definition 3, with $P_l = 50\%$.

5.3.3 Scalar concentration advected by an incompressible flow (ADR)

We consider the parametric semi-linear advection diffusion reaction equation in $d = 2$ dimensions, with $m = 3$ equations, rewritten in mixed form:

$$\begin{cases} \kappa^{-1}\sigma + \nabla u = 0, & \text{in } \Omega, \\ \nabla \cdot \sigma + \mathbf{v} \cdot \nabla u + u = f, & \text{in } \Omega, \\ \sigma \cdot \mathbf{n} = 0, & \text{on } \Gamma_N \cup \Gamma_{D,0}, \\ u = \sum_{i=1}^P \mu_i \chi_{I_i}, & \text{on } \Gamma_D, \end{cases} \quad (81)$$

where $\kappa = 0.05$ is fixed for this study,

$$\boldsymbol{\rho} = (\mu_1, \dots, \mu_P) \in \mathcal{P} \subset \mathbb{R}^P, \quad \mathcal{P} = \{\boldsymbol{\rho} \in \{0, 1\}^P \mid \mu_i = 1, \mu_j = 0, \forall j \in \{0, \dots, 99\} \setminus \{i\}\}, \quad (82)$$

and $\{\chi_{I_i}\}_{i=0}^{N_{\text{par}}}$ are the characteristic functions of the symmetric intervals $I_i = 0 \times [-i0.01 + 1.5, i0.01 + 2.5]$, with $N_{\text{par}} = 99$. The domain is shown in Figure 13. The advection velocity \mathbf{v} is obtained from the following incompressible Navier-Stokes equation at $t = 2\text{s}$:

$$\begin{cases} \partial_t \mathbf{v} + \mathbf{v} \cdot \nabla \mathbf{v} - \nu \Delta \mathbf{v} + \nabla p = \mathbf{0}, & \text{in } \Omega \\ \nabla \cdot \mathbf{v} = 0, & \text{in } \Omega \\ \mathbf{v} \times \mathbf{n} = 0, p = 0, & \text{on } \Gamma_N \\ \mathbf{v} = 0, & \text{on } \Gamma_{D,0} \\ \mathbf{v}(t = 0) = \mathbf{v}_b, & \text{on } \Gamma_D \end{cases} \quad (83)$$

with initial conditions on the boundary Γ_D , $\mathbf{v}_b = \mathbf{v}(x, y, t = 0) = (6y(4.1 - y)/4.1^2, 0) \in \mathbb{R}^2$ and $\nu \in \mathbb{R}$ such that the Reynolds number is $Re = 100$. The implementation is the one of step-35 of the tutorials of the deal.II library [6]. Homogeneous

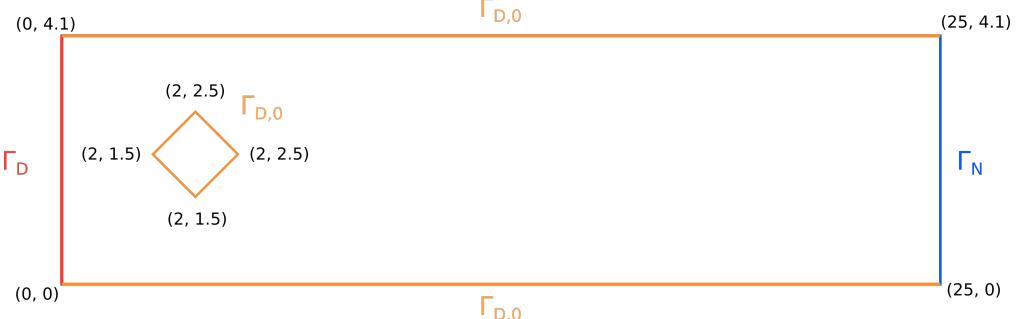


Figure 13: **ADR.** Computational domain of the advection diffusion reaction equation FS (81) and the incompressible Navier-Stokes equations (83). The boundary conditions specified for each system are reported in the text.

Neumann boundary on $\Gamma_N \cup \Gamma_{D,0}$ and Dirichlet non-homogeneous boundary conditions on Γ_D are applied with the boundary operator (30). A sample solution is shown in Figure 14 for $\mu_i = 0, i = 0, \dots, 98$ and $\mu_{99} = 1, \kappa = 0.05$. We remark that, for the moment, we consider only fixed values of $\kappa = 0.05$. For a convergence of ROMs to vanishing viscosity solutions with graph neural networks, see Section 6.

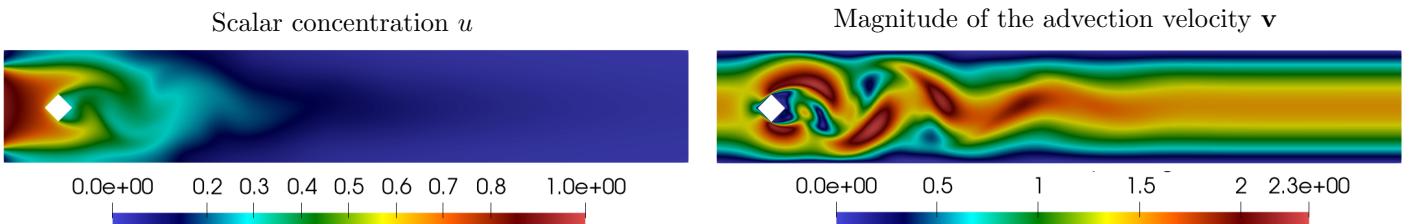


Figure 14: **ADR.** **Left:** scalar concentration u of the advection diffusion reaction equations (81), with $\mu_i = 0, i = 0, 98$ and $\mu_{99} = 1, \kappa = 0.05$. **Right:** advection velocity employed for the FS (81), obtained as the velocity \mathbf{v} from the INS (83) at $t = 2\text{s}$.

The FOM partitioned and DD-ROM repartitioned subdomains are shown in Figure 15. We choose the variance indicator to repartition the computational subdomain in two subset: 21% of the cells for the *low variance* part and 79% for the *high variance* part. With respect to the previous test cases, now it is evident the change in the order of magnitude of the local relative L^2 -reconstruction error in Figure 16, especially for the cellwise variance indicator I_{var} . We expect that lowering the local reduced dimension of the *low variance* repartitioned region will not affect sensibly the accuracy.

We use for the monodomain approach $r_{\Omega} = 5$ reduced basis as well as $r_{\Omega_i} = 5$ for $i = 1, \dots, K$ for the FOM partitioned subdomains. In the DD-ROM approach, we can use even $r_{\Omega_1} = 2$ and $r_{\Omega_2} = 5$ for the lower and higher variance subdomains, respectively, without affecting the error of the ROM solution, as we see in Figure 17. Indeed, the accuracy in terms of L^2 and energy norms is essentially identical for all approaches, even with so little number of basis functions for the DD-ROM one.

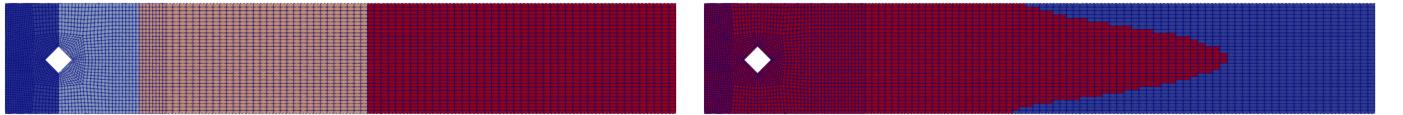


Figure 15: **ADR.** Domain of advection diffusion reaction equation. **Left:** computational subdomains partitioned in $K = 4$ subdomains by `petsc4py` inside `deal.II`. **Right:** DD-ROM repartition of the computational subdomain $k = 2$ with the cellwise variance indicator I_{var} , Definition 3: 21% of the cells belong to the *low variance* part, represented in blue, and the other 79% belong to the *high variance* part in red.

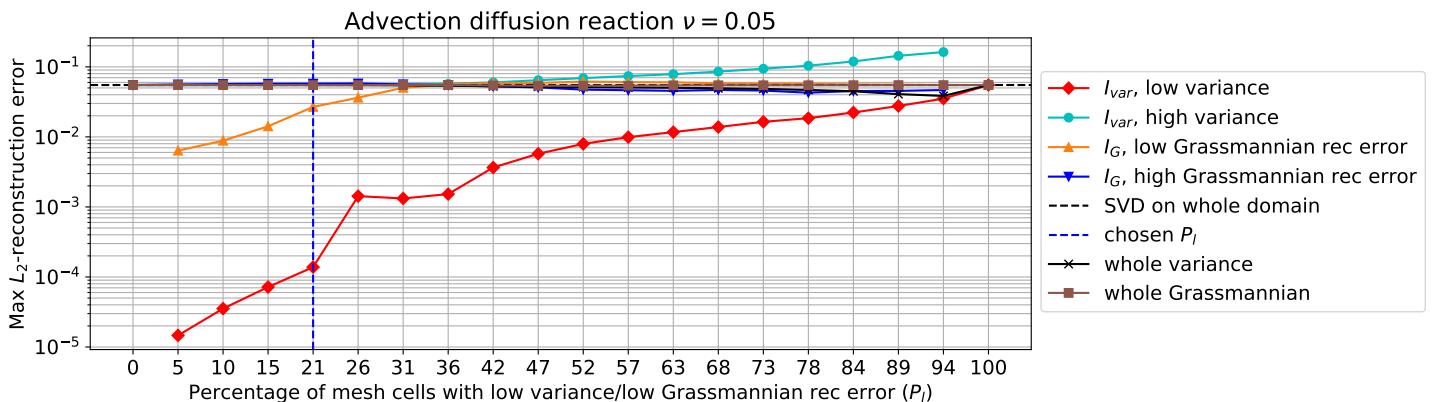


Figure 16: **ADR.** Local relative L^2 -reconstruction errors of the snapshots matrix of the advection diffusion reaction equation restricted to the two subdomains of the repartitioning performed with the indicator I_{var} (in red and light-blue), Definition 3, and I_G (in orange and blue), Definition 4. The relative L^2 -reconstruction error attained on the whole domain is shown in black for the indicator I_{var} and in brown for the indicator I_G . The local reduced dimensions used to evaluate the local reconstruction errors is $r_{\Omega_i} = 3$, $i = 1, 2$.

Again, we evaluate $n_{\text{train}} = 20$ training full-order solutions and $n_{\text{test}} = 80$ test full-order solutions, corresponding to the parameter choices $\mu_i = 1$ and $\mu_{\bar{i}} = 0$, for $i = 0, \dots, 99$, with fixed viscosity $\kappa = 0.05$, where \bar{i} represents all the indices in $\{0, \dots, 99\}$ except from i . So, the training snapshots correspond to $i = 0, 5, 10, \dots, 95$. For these studies we have fixed the local dimensions to $r_{\Omega_i} = 5$, $i = 1, \dots, K$ for $K = 4$, $r_{\Omega} = 5$ for the whole computational domain and $r_{\Omega_1} = 2$, $r_{\Omega_2} = 5$ for the repartitioned case with $k = 2$, as mentioned.

In Table 3, we list the computational times and speedups for a simulation with the different methods.

6 Graph Neural Networks approximating Vanishing Viscosity solutions

In this section, we want to highlight how the well-known concept of vanishing viscosity solutions can be related to FS. In hyperbolic problems, the uniqueness of the weak solution is not guaranteed, already for very simple problems, e.g. inviscid Burgers' equations. In order to filter out the physically relevant solution, the concept of vanishing viscosity solution has been introduced, *inter alia* [37], and, consequently, vanishing viscosity methods have been developed, e.g. [26, 63].

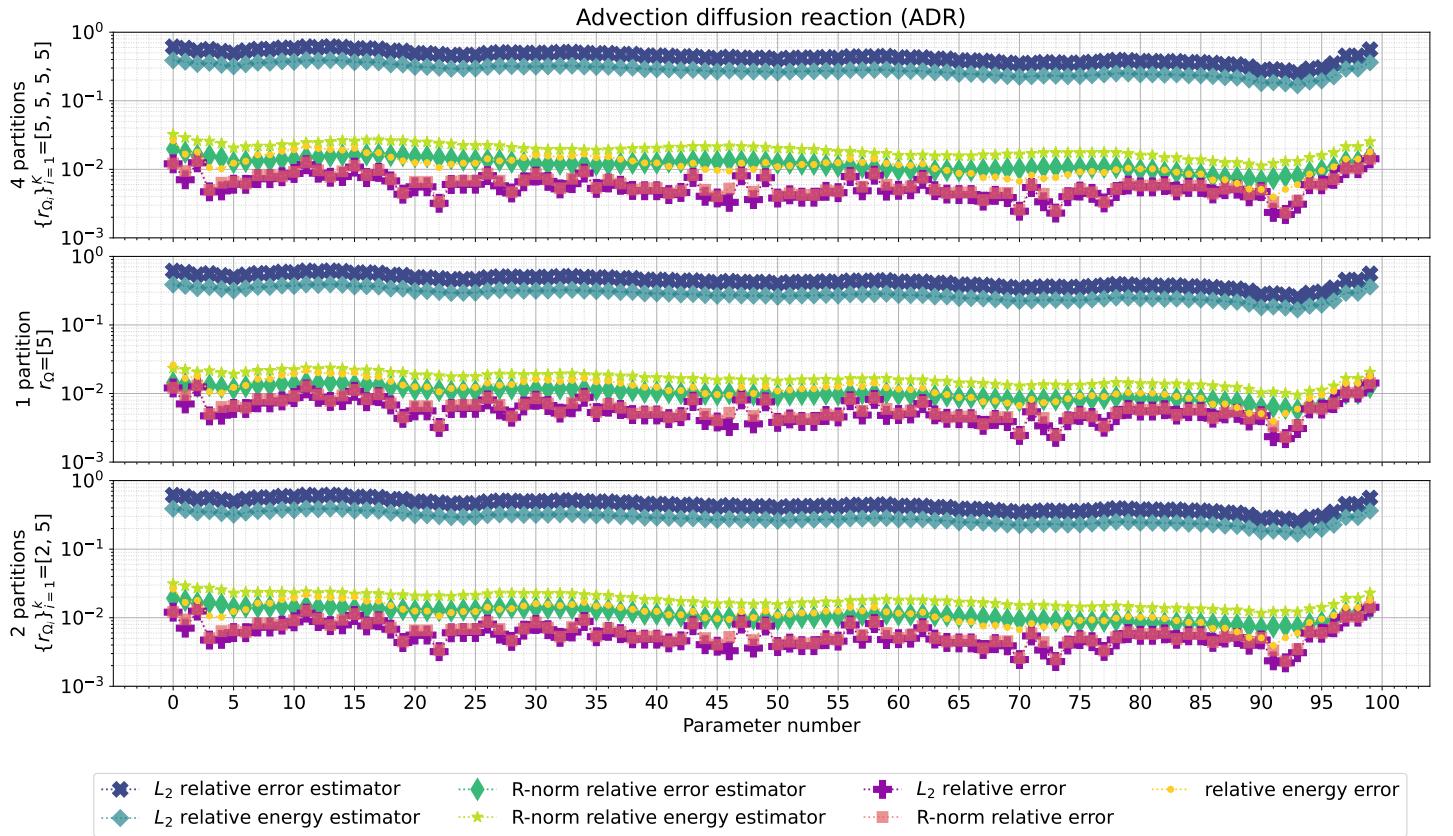


Figure 17: **ADR.** Errors and estimators for advection diffusion reaction equation corresponding to the $n_{\text{train}}=20$ uniformly sampled train snapshots corresponding to the abscissae $0, 5, 10, \dots, 95$, and $n_{\text{test}}=80$ uniformly sampled test snapshots, corresponding to the other abscissae. The reduced dimensions of the ROMs are $\{r_{\Omega_i}\}_{i=1}^K = [5, 5, 5, 5]$ for $K = 4$ partitions, $r_{\Omega} = 5$ for $k = 1$ partition, and $\{r_{\Omega_i}\}_{i=1}^k = [2, 5]$ for $k = 2$ partitions. For the case $k = 2$ we employed the cellwise variance indicator I_G , Definition 3, with $P_l = 21\%$.

Table 3: **ADR.** Average computational times and speedups for ROM and DD-ROM approaches for Maxwell equations. The speedup is computed as the FOM computational time over the ROM one. The FOM runs in parallel with 4 cores, so “FOM time” refers to wallclock time.

FOM		ROM			DD-ROM		
N_h	time	r	time	speedup	r_i	time	speedup
131328	3.243 [s]	5	79.112 [μ s]	~ 40992	[5, 5, 5, 5]	59.912 [μ s]	~ 54129

We will consider the topic of vanishing viscosity solutions from the different perspective of model order reduction. It is known that slow decaying Kolmogorov n-width solution manifolds result in ineffective linear reduced order models. The origin of this problem rests theoretically on the regularity of the parameter to solution map [18, 19], and with less generality on the nature of some PDEs (e.g. advection dominated PDEs, nonlinearities, complex dynamics), on the size of the parameter space, and on the smoothness of the parametric initial data or parametric boundary conditions [5], mainly. A possible way to obtain more approximable solution manifolds is through regularization or filtering [87, 84], e.g. adding artificial viscosity. Heuristically, the objective is to smoothen out the parametric solutions of the PDEs, for example removing sharp edges, local features, complex patterns, with the aim of designing more efficient ROMs for the filtered solution manifolds. Then, the linear ROMs will be applied to different levels of *regularization*, still remaining in the regime where they have good approximation properties. Finally, the original (vanishing viscosity) solutions will be recovered with a regression method from the succession of filtered linear ROMs. This is realized without the need to directly reduce with a linear reduced manifold the original solution manifold, thus avoiding the problem of its approximability with a linear subspace and the slow Kolmogorov n-width decay.

In our case, we consider regularization by viscosity levels: the vanishing viscosity solutions u_ν with viscosity $0 \leq \nu \ll 1$, will be recovered as the limit $\lim_{i \rightarrow \infty} u_{\nu_i} = u_\nu$ of a potentially infinite succession of viscosity levels $\{\nu_i\}_{i=0}^\infty$, $\nu_0 > \nu_1 > \dots > 0$, each associated to its efficient reduced order model. In practice, $\{\nu_i\}_{i=0}^\infty \approx \{\nu_i\}_{i=0}^q$, where q is the number of additional viscosity ROMs. It is clear the connection with multi-fidelity and super-resolution methods [35, 56]. The rationale of the approach is supported by the proofs of convergence to vanishing viscosity solutions of hyperbolic PDEs under various hypotheses [64, 57, 25, 38].

The framework is general and can be applied in particular to FS. We will achieve this for the advection–diffusion–reaction problem changing the viscosity constant $\mathbb{R} \ni \kappa > 0$ in (81). While this choice is specific for the model we are considering, a more general approach could consist in adding a viscous dissipative term to the generic FS obtaining another FS:

$$\begin{cases} Au = f + \kappa \Delta \mathbf{u}, & \text{in } \Omega \\ (\mathcal{D} - \mathcal{M})(u - g) = 0, & \text{on } \partial\Omega \end{cases} \rightarrow \begin{cases} \kappa^{-1} \sigma + \nabla u = 0 \\ \nabla \cdot \sigma + Au = f \\ (\mathcal{D} - \mathcal{M})(u - g) = 0, \end{cases}, \quad \text{in } \Omega \quad (84)$$

recalling that the additional degrees of freedom are needed only for the high viscosity ROMs and FOMs (to collect the snapshots) and not the full-order vanishing viscosity solutions. This is only an example of how the procedure could be applied to any FS. In fact, the methodology is not designed specifically for FS.

The overhead of the methodology is related to the evaluation of the snapshots, the assembling of each level of viscosity $\{\nu_i\}_{i=0}^q$, and the computational costs of the regression method. We remark that the full matrices of the affine decomposition of each $\{ROM_{\nu_i}\}_{i=0}^q$ are the same. This is the price necessary to tackle the realization of reduced order models of parametric PDEs affected by a slow Kolmogorov n-width decay with our approach.

With respect to standard techniques for nonlinear manifold approximation, the proposed one is more interpretable as a mathematical limit of a succession of solutions to the vanishing viscosity one. Moreover, it has a faster training stage relying on the efficiency of the $\{ROM_{\nu_i}\}_{i=0}^q$. To the authors' knowledge, cheap analytical ways to obtain the vanishing viscosity solution from a finite succession of high viscosity ones are not available, so we will rely on data-driven regression methods.

6.1 Graph neural networks augmented with differential operators

Generally, machine learning (ML) architectures are employed in surrogate modelling to approximate nonlinear solution manifolds, otherwise linear subspaces are always preferred. The literature is vast on the subject and there are many frameworks that develop surrogate models with ML architectures. They promise to define data-driven reduced order models that infer solutions for new unseen parameters provided that there are enough data to train such architectures. This depends crucially on the choice of the encoding and inductive biases employed to represent the involved datasets: the training computational time and the amount of training data can change drastically.

On this matter, convolutional autoencoders (CNN) are one of the most efficient architectures to approximate nonlinear solution manifolds [59] for data structured on Cartesian grids, mainly thanks to their shift-equivariance property. For fields on unstructured meshes the natural choice are Graph neural networks (GNNs). Since their employment, GNNs architectures from the ML community have been enriched with physical inductive biases and other tools from numerical analysis. We want to test one of the first implementations and modifications of GNNs [80]. We also want to remark that in the literature, there are still very few test cases of ROMs that employ GNNs with more than ≥ 50000 degrees of freedom. The difficulty arises when the training is performed on large meshes, thus the need for tailored approaches.

The majority of GNNs employed for surrogate modelling are included in autoencoders [33, 67] or are directly parametrized to infer the unseen solution with a forward evaluation. These architectures may become heavy, especially for non-academic test cases. One way to tackle the problem of parametric model order reduction of slow Kolmogorov n-width solution manifolds is to employ GNNs only to recover the high-fidelity solution in a multi-fidelity setting, through super-resolution. Since efficient

ROMs are employed to obtain the lower levels of fidelity (high viscosity solutions in our case), the solution manifold dimension reduction is performed only at those levels, avoiding the costly and heavy in memory training of autoencoders of GNNs.

We describe the implementation of augmented GNNs as in [80], with the difference that we need to train only a map from a collection of DD-ROMs solutions to the full-order vanishing viscosity solution, and not an autoencoder with pooling and unpooling layers to perform dimension reduction. The GNN we will employ is rather thin with respect to autoencoder GNNs used to perform dimension reduction. Its details are reported in Table 4.

We represent with

$$\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{W}), \quad \mathcal{V} \in \mathbb{R}^{n_{\text{nodes}} \times f}, \quad \mathcal{E} \in \mathbb{N}^{n_{\text{edges}} \times 2}, \quad \mathcal{W} \in \mathbb{R}^{n_{\text{attr}} \times d}, \quad (85)$$

a graph with node features \mathcal{V} , edges \mathcal{E} and edge attributes \mathcal{W} . The number f represents the nodal features dimension. We denote with $\mathbf{e}_{ij} = (i, j) \in \mathbb{N}^2$ the edge between the nodes $\mathbf{n}_i, \mathbf{n}_j \in \mathbb{R}^f$: \mathbf{e}_{ij} corresponds to a row of \mathcal{E} , and $\mathbf{n}_i, \mathbf{n}_j$ correspond to the i -th and j -th rows of \mathcal{V} , for $i, j = 1, \dots, n_{\text{nodes}}$. Similarly, $\boldsymbol{\omega}_{ij}$ represents the edge attributes of edge \mathbf{e}_{ij} . We have $n_{\text{edges}} = n_{\text{attr}}$. For their efficiency, GNNs rely on a message passing scheme composed of propagation and aggregation steps. Supposing that the graph is sparsely connected their implementation is efficient.

When the graph is supported on a mesh, it is natural to consider the generalized support points of finite element spaces as nodes of the graph and the sparsity pattern of the linear system associated to the numerical model as the adjacency matrix of the graph. We employ only Lagrangian nodal basis of discontinuous finite element spaces, but the framework can be applied to more general finite element spaces. As edge attributes $\boldsymbol{\omega}_{ij}$, we will employ the difference $\boldsymbol{\omega}_{ij} = \mathbf{x}_i - \mathbf{x}_j \in \mathbb{R}^d$ between the corresponding spatial coordinates associated to the nodes $\mathbf{n}_i, \mathbf{n}_j \in \mathbb{R}^f$. The nodes adjacent to node \mathbf{n}_i are represented with the set $\mathcal{N}_{\text{neigh}(i)}$ for all $i = 1, \dots, n_{\text{nodes}}$.

We consider only the two following types of GNN layers: a continuous kernel-based convolutional operator l_{NNconv} [36, 77] and the GraphSAGE operator l_{SAGEconv} [40],

$$\mathcal{V}_{\text{out}} = l_{\text{NNconv}}(\mathcal{V}_{\text{inp}}, \mathcal{E}, \mathcal{W}) = \mathcal{V}_{\text{inp}} W_3 + \text{Avg}_1(\mathcal{V}_{\text{inp}}, h(\mathcal{W})) + \mathbf{b}_3, \quad h(\mathcal{W}) = \text{ReLU}(\mathcal{W} W_1 + \mathbf{b}_1) W_2 + \mathbf{b}_2, \quad (86)$$

$$\mathcal{V}_{\text{out}} = l_{\text{SAGEconv}}(\mathcal{V}_{\text{inp}}, \mathcal{E}, \mathcal{W}) = \mathcal{V}_{\text{inp}} W_6 + \text{Avg}_2(\text{ReLU}(\mathcal{V}_{\text{inp}} W_4 + \mathbf{b}_4)) W_5 + \mathbf{b}_5, \quad (87)$$

with weight matrices dimensions,

$$W_1 \in \mathbb{R}^{2 \times l}, \quad W_2 \in \mathbb{R}^{l \times (f_{\text{inp}} \times f_{\text{out}})}, \quad W_3 \in \mathbb{R}^{f_{\text{inp}} \times f_{\text{out}}}, \quad W_4 \in \mathbb{R}^{f_{\text{inp}} \times f_{\text{inp}}}, \quad W_5, W_6 \in \mathbb{R}^{f_{\text{inp}} \times f_{\text{out}}}, \quad (88)$$

$$\mathbf{b}_1 \in \mathbb{R}^l, \quad \mathbf{b}_2 \in \mathbb{R}^{(f_{\text{inp}} \times f_{\text{out}})}, \quad \mathbf{b}_3, \mathbf{b}_4 \in \mathbb{R}^{f_{\text{out}}}, \quad \mathbf{b}_5 \in \mathbb{R}^{f_{\text{inp}}}, \quad (89)$$

$$h(\mathcal{W}) \in \mathbb{R}^{n_{\text{edges}} \times (f_{\text{inp}} \times f_{\text{out}})}, \quad \mathcal{W} = \{W_s^h\}_{s=1}^{n_{\text{edges}}}, \quad W_s^h \in \mathbb{R}^{f_{\text{inp},s} \times f_{\text{out},s}}, \quad \forall s = 1, \dots, n_{\text{edges}}, \quad (90)$$

with the following average operators used as aggregation operators,

$$(\text{Avg}_1(\mathcal{V}, \{W_s^h\}_{s=1}^{n_{\text{edges}}}))_i = \frac{1}{|\mathcal{N}_{\text{neigh}(i)}|} \sum_{s \in \mathcal{N}_{\text{neigh}(i)}} W_s^h \mathbf{n}_s, \quad (\text{Avg}_2(\mathcal{V}))_i = \frac{1}{|\mathcal{N}_{\text{neigh}(i)}|} \sum_{s \in \mathcal{N}_{\text{neigh}(i)}} \mathbf{n}_s, \quad \forall i = 1, \dots, n_{\text{nodes}}, \quad (91)$$

where $\mathcal{V}_{\text{inp}} \in \mathbb{R}^{n_{\text{nodes}} \times f_{\text{inp}}}$, $\mathcal{V}_{\text{out}} \in \mathbb{R}^{n_{\text{nodes}} \times f_{\text{out}}}$ are the input and output nodes with feature dimensions $f_{\text{inp}}, f_{\text{out}}$. We remark that, differently from graph neural networks with heterogeneous layers, i.e., with changing mesh structure between different layers, in this network the edges \mathcal{E} and edge attributes \mathcal{W} are kept fixed, only the node features change. The feed-forward neural network $h : \mathbb{R}^{n_{\text{edges}} \times d} \rightarrow \mathbb{R}^{f_{\text{inp},s} \times f_{\text{out},s}}$ defines a weight matrix $W_s^h \in \mathbb{R}^{f_{\text{inp},s} \times f_{\text{out},s}}$ for each edge $s = 1, \dots, n_{\text{edges}}$. The number l is the hidden layer dimension of h .

The aggregation operators are defined from the edges \mathcal{E} that are related to the sparsity pattern of the linear system of the numerical model. So, the aggregation is performed on the stencils of the numerical scheme chosen for every layer of the GNN architecture in Table 4. Many variants are possible, in particular, we do not employ pooling and unpooling layers to move from different meshes: we always consider the same adapted mesh.

Since our GNNs work on the nodal features, a good strategy is to augment their dimensions as proposed in [80]. In fact, in the majority of applications of GNNs for physical models the input features dimensions is the dimension of the physical fields considered and it is usually very small. Considering FS, the fields' dimension is m . To augment the input features, we will filter them with some differential operators discretized on the same mesh in which the GNN is supported. We consider the following differential operators

$$\Delta : V_h(\Omega) \rightarrow V_h(\Omega), \quad (\text{Laplace operator}) \quad (92)$$

$$\mathbf{v} \cdot \nabla : V_h(\Omega) \rightarrow V_h(\Omega), \quad (\text{Advection operator}) \quad (93)$$

$$\nabla_x : V_h(\Omega) \rightarrow V_h(\Omega), \quad (\text{Gradient x-component}) \quad (94)$$

$$\nabla_y : V_h(\Omega) \rightarrow V_h(\Omega), \quad (\text{Gradient y-component}) \quad (95)$$

for a total of four possible feature augmentation operators, where, in our case, \mathbf{v} is the advection velocity from the incompressible Navier-Stokes equations (83). We employ the representation of the previous differential operators with respect to

the polynomial basis of Lagrangian shape functions, so they act on the vectors of nodal evaluations in \mathbb{R}^{N_h} . As in [80], we consider three sets of possible augmentations:

$$\mathcal{O}_1 = \{\mathbb{I}_{N_h}, \Delta, \mathbf{v} \cdot \nabla, \nabla_x, \nabla_y\}, \quad (96)$$

$$\mathcal{O}_2 = \{\mathbb{I}_{N_h}, \nabla_x, \nabla_y\} \quad (97)$$

$$\mathcal{O}_3 = \{\mathbb{I}_{N_h}\} \quad (98)$$

where \mathbb{I}_{N_h} is the identity matrix in \mathbb{R}^{N_h} , $|\mathcal{O}_1| = 5 = n_{\text{aug}}$, $|\mathcal{O}_2| = 3 = n_{\text{aug}}$ and $|\mathcal{O}_3| = 1 = n_{\text{aug}}$. We will reconstruct only the scalar concentration u with the GNN, so, in our case, the field dimension is 1, which is the output dimension. The input dimension depends on the number of high viscosity DD-ROMs employed that we denote with q . Given a single parametric instance $\rho \in \mathbb{R}^P$ the associated solutions of $\{\text{D-ROM}_{\kappa_i}\}_{i=1}^q$ are $\{\mathbf{u}^{\text{RB}}(\rho_i)\}_{i=1}^q$.

We divide the snapshots $\{\mathbf{u}^{\text{RB}}(\rho_i)\}_{i=1}^{n_{\text{train}}+n_{\text{test}}}$ in training $\{\mathbf{u}^{\text{RB}}(\rho_i)\}_{i \in I_{\text{train}}}$ and test snapshots $\{\mathbf{u}^{\text{RB}}(\rho_i)\}_{i \in I_{\text{test}}}$, with $|I_{\text{train}}| = n_{\text{train}}$ and $|I_{\text{test}}| = n_{\text{test}}$. We have decided to encode the reconstruction of the vanishing viscosity solution \mathbf{u}_{q+1} learning the difference $\mathbf{u}_{q+1}(\rho) - \mathbf{u}_q^{\text{RB}}(\rho) - \bar{\mathbf{u}}_{q+1 \text{train}}$ with the mesh-supported-augmented GNN (MSA-GNN) described in Table 4:

$$\mathbf{u}_{q+1}(\rho) = \text{ReLU} \left(\mathbf{u}_q^{\text{RB}}(\rho) + \text{MSA-GNN} \left(\{O_a \{\{\mathbf{u}_i^{\text{RB}}(\rho)\}_{i=1}^q, \{\mathbf{u}_i^{\text{RB}}(\rho) - \mathbf{u}_{i-1}^{\text{RB}}(\rho)\}_{i=1}^{q-1}\}\}_{a=2}^{n_{\text{aug}}} \right) + \bar{\mathbf{u}}_{q+1 \text{train}} \right), \quad (99)$$

where $\bar{\mathbf{u}}_{q+1 \text{train}} = \frac{1}{n_{\text{train}}} \sum_{i=1}^{n_{\text{train}}} \mathbf{u}_{q+1}(\rho_i)$. Learning the difference instead of the solution itself helps in getting more informative features. The input dimension is therefore $3n_{\text{aug}} = 15$ for \mathcal{O}_1 and $3n_{\text{aug}} = 9$ for \mathcal{O}_2 .

Table 4: Mesh supported augmented GNN

Net	Weights $[f_{\text{inp}}, f_{\text{out}}]$	Aggregation	Activation
Input NNConv	[$3n_{\text{aug}}$, 18]	Avg ₁	ReLU
SAGEconv	[18, 21]	Avg ₂	ReLU
SAGEconv	[21, 24]	Avg ₂	ReLU
SAGEconv	[24, 27]	Avg ₂	ReLU
SAGEconv	[27, 30]	Avg ₂	ReLU
Output NNConv	[30, 1]	Avg ₁	-

NNConvFilters	First Layer $[2, l]$	Activation	Second Layer $[l, f_{\text{inp}}, f_{\text{out}}]$
Input NNConv	[2, 12]	ReLU	[12, $3n_{\text{aug}} \cdot 18$]
Output NNConv	[2, 8]	ReLU	[8, 30]

6.2 Decomposable ROMs approximating vanishing viscosity (VV) solution through GNNs

In this section, we test the proposed multi-fidelity approach that reconstruct the lowest viscosity level with the GNN. We consider the FS (81), with three levels of viscosity, from highest to lowest: $\kappa_1 = 0.05$, $\kappa_2 = 0.01$ and $\kappa_3 = 0.0005$. We want to build a surrogate model that efficiently predicts the parametric solutions of the FS (81) for unseen values of $\rho \in \mathbb{P}$ with fixed viscosity $\kappa_3 = 0.0005$. These solutions will be referred to as *vanishing viscosity* solutions. The other two viscosity levels are employed to build the D-ROM $_{\kappa_1}$ and D-ROM $_{\kappa_2}$ with viscosities $\kappa_1 = 0.05$ and $\kappa_2 = 0.01$, respectively. The parametrization affects the inflow boundary condition and is the same as the one described in section 5.3.3, see equation (82). We also employ the same number of training 20 and test 80 parameters.

The DD-ROMs provided for $\kappa_1 = 0.05$ and $\kappa_2 = 0.01$ can be efficiently designed with reduced dimensions $\{r_{\Omega_i}\}_{i=1}^K [5, 5, 5, 5]$. To further reduce the cost, we employ an even coarser mesh for ROM $_{\kappa_1}$ and ROM $_{\kappa_2}$ and a finer mesh for the vanishing viscosity solutions. The former is represented on the left of Figure 18, the latter on the right. The degrees of freedom related to the coarse mesh are 43776, while the ones on the fine one are 175104.

For the training of the GNN we use the open source software library PyTorch Geometric [32]. The employment of efficient samplers that partition the graphs on which the training set is supported is crucial to lower the otherwise heavy memory burden [40]. We preferred samplers that partition the mesh with METIS [54] as it is often employed in this context. We decided to train the GNN with early stopping at 50 epochs as our focus is also in the reduction of the training time of the NN architectures used for model order reduction. It corresponds on average to less than 60 minutes of training time. The batch size is 100 and we clustered the whole domain in 100000 subgraphs in order to fit the batches in our limited GPU memory. Each augmentation strategy and additional fidelity level, do not affect the whole training time as they only

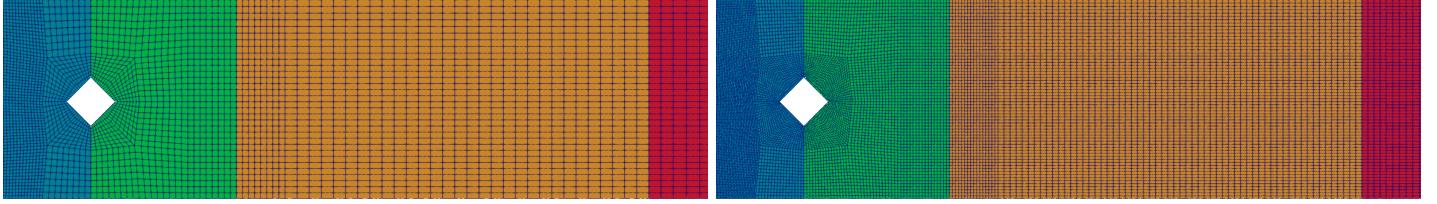


Figure 18: **VV.** Left part of the computational domain partitioned in 4 for distributed parallelism: coarse mesh (**left**), fine mesh (**right**). The solution with viscosity $\kappa \in \{0.05, 0.01\}$ are evaluated on the coarse mesh with 4868 cells and 43776 dofs, those with $\kappa = 0.0005$ on the finer with 19456 cells and 175104 dofs.

increase the dimension of the input features from a minimum of 1 (1 fidelity, no augmentation) to a maximum of 15 (all augmentations \mathcal{O}_3 , 2 fidelities). As optimizer we use ADAM [55] stochastic optimizer. Every architecture is trained on a single GPU NVIDIA Quadro RTX 4000.

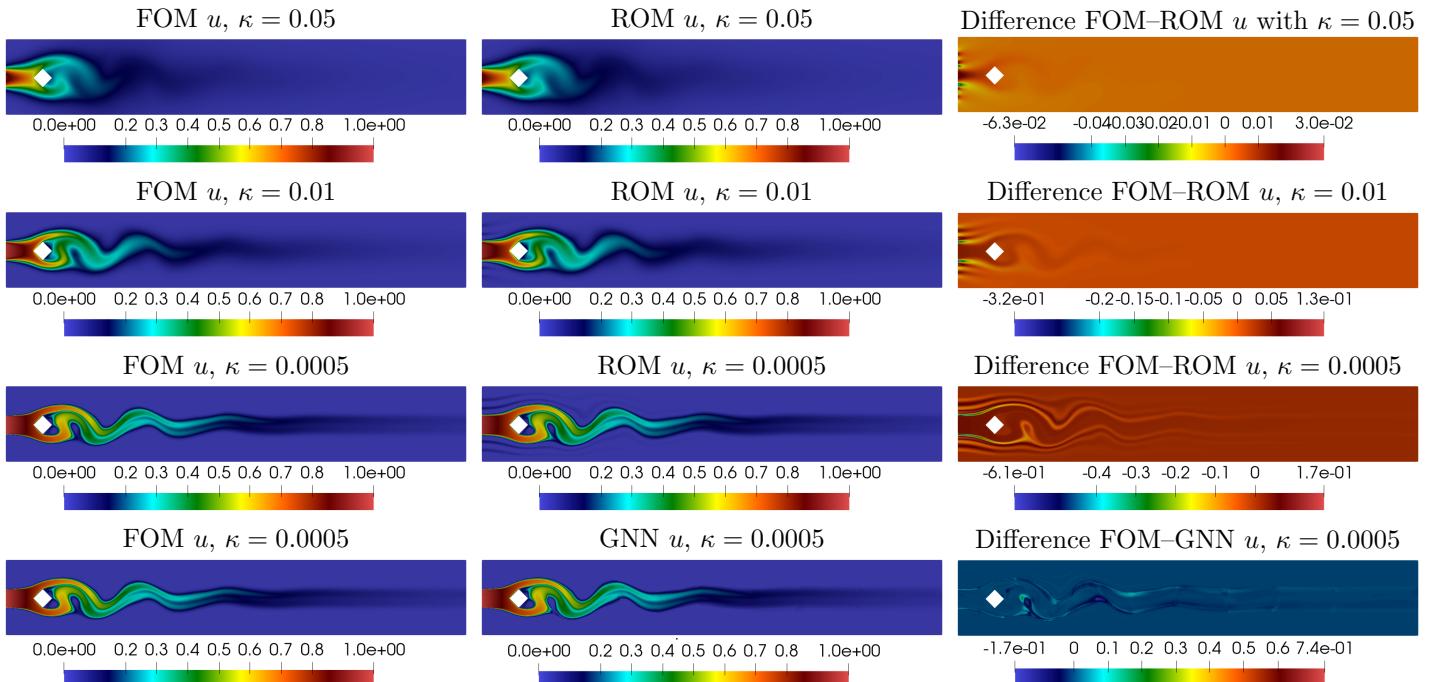


Figure 19: **VV.** Scalar concentration advected by incompressible flow for $i = 0$. Comparison of ROM approach at different viscosity levels $\kappa \in \{0.05, 0.01, 0.0005\}$ and GNN for $\kappa = 0.0005$. FOMs on the left, reduced solution at the center and error on the right.

Figures 19, 20 and 21 show the results of the algorithm for parameters with index $i \in \{0, 50, 99\}$. In particular, we show on the left columns the FOM simulations, in the center column the ROM simulations and the error in the right column. Moreover, in the different rows, we have different viscosity levels. The first three rows use the classical DD-ROM approach. We can immediately see that the vanishing viscosity $\kappa = \kappa_3 = 0.0005$ level shows strong numerical oscillations along the whole solution, which are not present in the FOM method. This phenomenon is observable also for higher viscosity levels but it is less pronounced and concentrated on the left of the domain, where the discontinuity are imposed as boundary conditions (see error plots). Finally, in the last row, we show the results of the GNN approach, which uses the first two viscosity levels to predict the vanishing viscosity one. Contrary the DD-ROM, we do not observe many numerical oscillations in the reduced solutions and they are much more physically meaningful. Thinking about extending this approach for more complicated problems, as Euler's equations, one could guarantee the presence of the correct amount of shocks and the right location or maintaining the positivity of density and pressure close to discontinuities.

In Figure 22, we show a quantitative measure of the error of the reduced approaches presented in terms of relative L^2 error. Overall, we can immediately see that the new GNN approach can always reach errors of the order of 1 – 2% for the

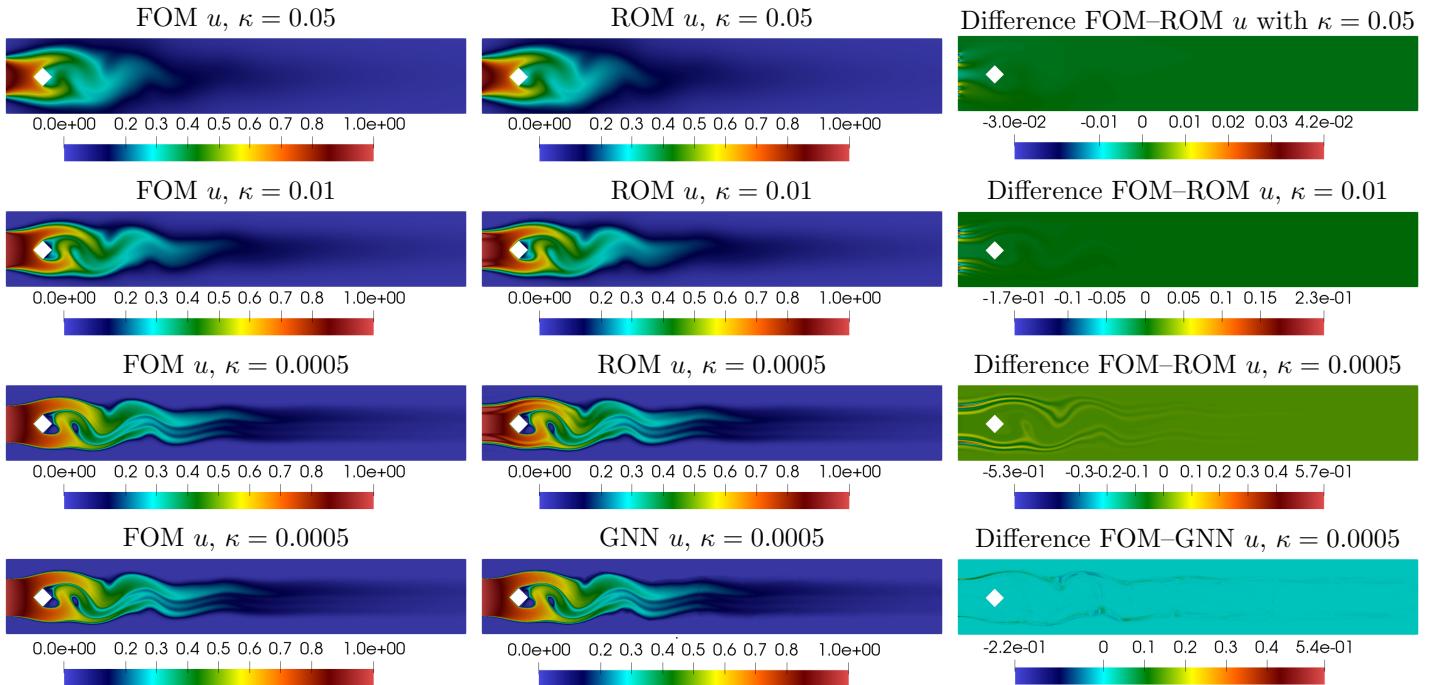


Figure 20: **VV.** Scalar concentration advected by incompressible flow for $i = 50$. Comparison of ROM approach at different viscosity levels $\kappa \in \{0.05, 0.01, 0.0005\}$ and GNN for $\kappa = 0.0005$. FOMs on the left, reduced solution at the center and error on the right.

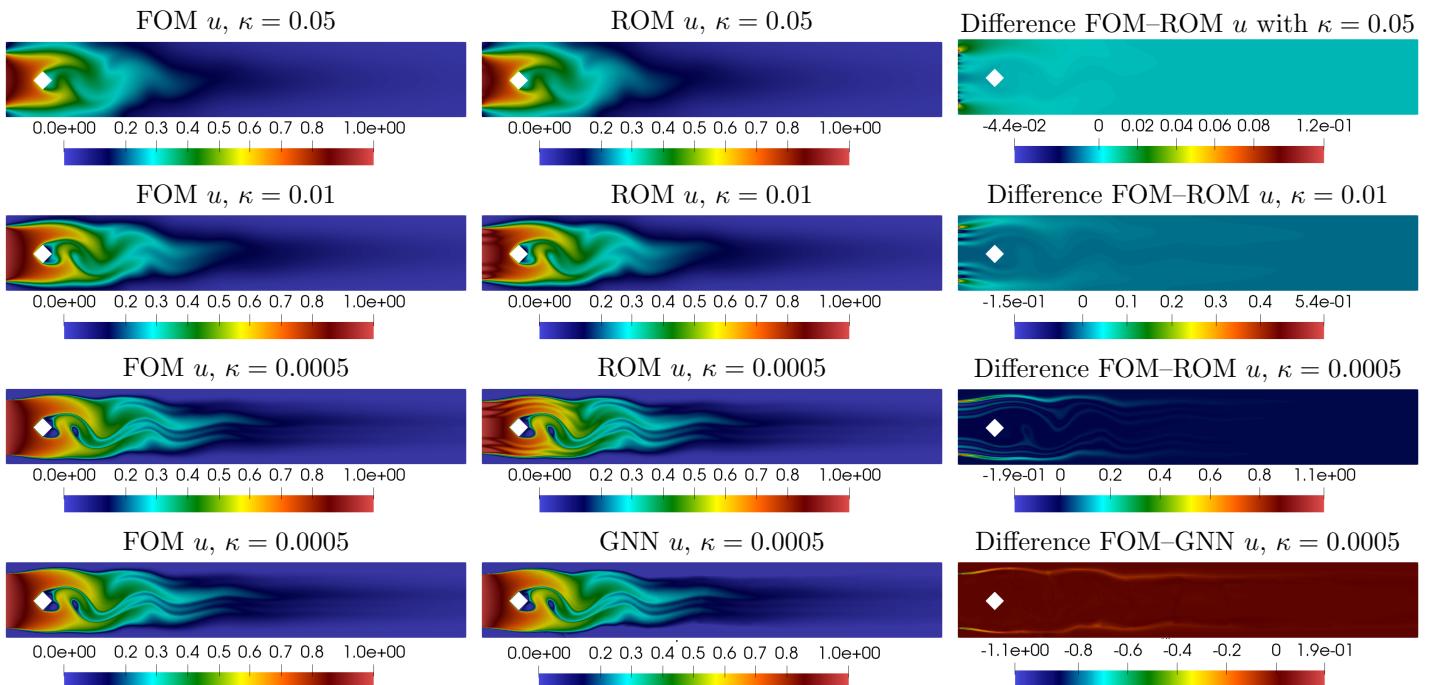


Figure 21: **VV.** Scalar concentration advected by incompressible flow for $i = 99$. Comparison of ROM approach at different viscosity levels $\kappa \in \{0.05, 0.01, 0.0005\}$ and GNN for $\kappa = 0.0005$. FOMs on the left, reduced solution at the center and error on the right.

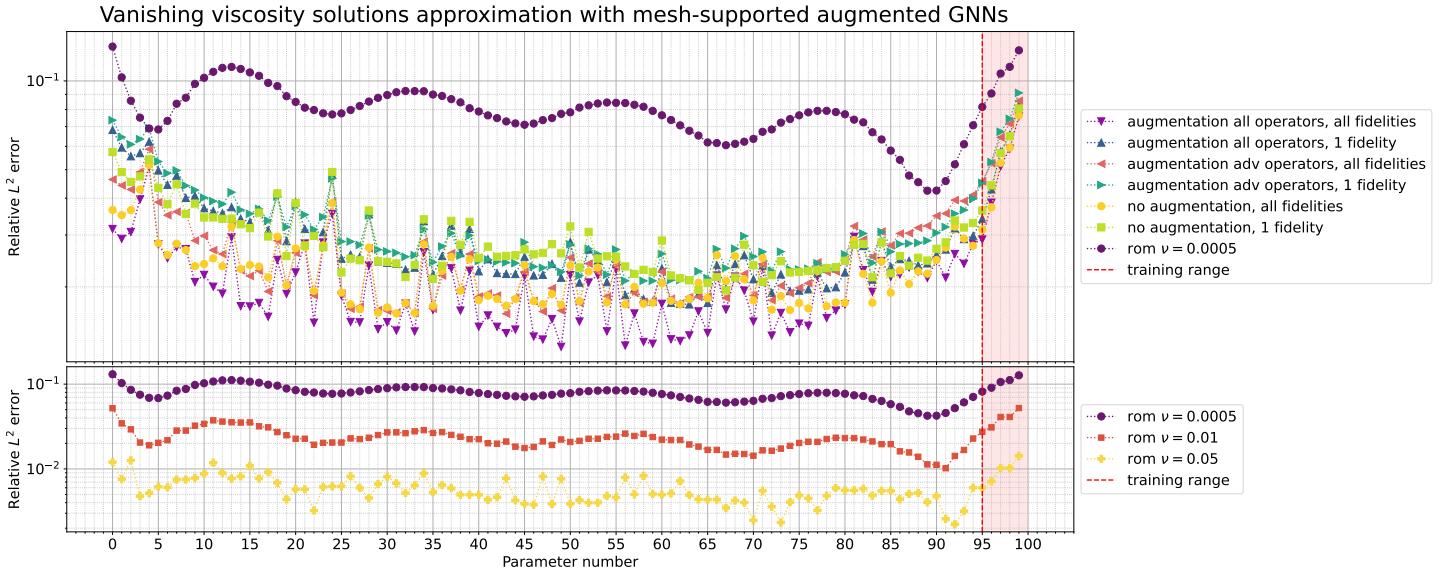


Figure 22: **VV.** Relative errors for the scalar conservation advected by incompressible flow problem. The parameters corresponding to the snapshots used for the GNNs and DD-ROMs training correspond to the abscissae $0, 5, 10, \dots, 95$ the rest are test parameters. The dashed red background highlights the extrapolation range. **Top:** errors on train and test set with different GNN approaches given by the three augmentation $\mathcal{O}_1, \mathcal{O}_2$ and \mathcal{O}_3 and by using either 1 viscosity level (1 fidelity) or 2 (all fidelities) and errors for DD-ROM with the same viscosity level $\nu = 0.0005$. **Bottom:** errors for DD-ROM approaches at different viscosity levels. The reduced dimensions of the ROMs are $\{r_{\Omega_i}\}_{i=1}^K = [5, 5, 5, 5]$ with $K = 4$ partitions.

vanishing viscosity solutions, with few peaks in the extrapolatory regime of 8%, while the classical DD-ROM on the vanishing viscosity solutions perform worse, with errors around 6-10%. On the other hand, the DD-ROM for higher viscosity levels have lower errors around 3% for κ_2 and 0.5% for κ_1 , hence, they are still reliably representing those solutions.

On the different GNN approaches, in Figure 22 at the top we compare the different augmentations $\mathcal{O}_1, \mathcal{O}_2$ and \mathcal{O}_3 and how many levels of viscosity we keep into considerations to derive the vanishing viscosity solution. The usage of multiple fidelity levels (two viscosity levels) is a great improvement for all the augmentations proposed and it can make gain a factor of 2 in terms of accuracy. There are slight differences with the used augmentations and, in particular, we observe that the \mathcal{O}_1 augmentation, with all operators, guarantee better performance, while there are no appreciable differences between \mathcal{O}_2 and \mathcal{O}_3 . Clearly, one could come up with many other augmentation possibilities choosing more operators, but at a cost of increasing the dimensions of the GNN and the offline training costs. We believe that all the presented options already perform much better with respect to classical approaches and can already be used without further changes.

In Table 5, we compare the computational times necessary to compute the FOM solutions, the DD-ROM ones, the training time for the GNN and the online costs of the GNN. As mentioned before, we employ only one GPU NVIDIA Quadro

Table 5: **VV.** Computational costs for scalar advected by a incompressible flow problem with GNNs approximating vanishing viscosity solutions (VV). The speedup is computed as the FOMs computational time over the ROM one. The speedup of the GNN is with respect to the FOM with viscosity $\nu = 0.0005$. The FOM runs in parallel with $K = 4$ cores as the DD-ROMs, so “FOM time” and “DD-ROM time” refers to wallclock time. Regarding the GNN results, “Single forward GNN online time” refers to a single online evaluation while “Total online time” refers to the evaluation of the 100 training and test snapshots altogether with only two separate GNN forward evaluations with batches of 50 inputs each. The speedup is evaluated as “FOM time” over “Total online time” divided by 100.

κ	FOM		DD-ROM			
	N_h	time	r_i	time	speedup	mean L^2 error
0.05	43776	3.243 [s]	[5, 5, 5, 5]	59.912 [μ s]	54129	0.00595
0.01	43776	3.236 [s]	[5, 5, 5, 5]	79.798 [μ s]	40552	0.0235
0.0005	175104	9.668 [s]	[5, 5, 5, 5]	95.844 [μ s]	100872	0.0796

κ	GNN training time	Single forward GNN online time	Total online time	GNN speedup	mean L^2 error
0.0005	≤ 60 [min]	2.661 [s]	17.166 [s]	~ 56	0.0217

RTX 4000 with 8GB of memory. Typical GNNs applications that involve autoencoders to perform nonlinear dimension reduction are much heavier. The training time of the GNNs for the different choice of augmentation operators vary between 48 minutes and 60 minutes approximately. We believe that in the near future more optimized implementations will reduce the training costs of GNNs. The computational time of the evaluation of a single forward of the GNN is on average 2.661 seconds but vectorization ensures the evaluation of multiple online solutions altogether: with our limited memory budget we could predict all the 100 training and test snapshots with just 2 batches of 50 stacked inputs each. The “Total online time” computed as previously described is 17.166 seconds that is 171.66 milliseconds per online solution with a speedup of around 56 with respect to the 9.668 seconds for the FOM.

Although the speedup for the GNN simulations are not as remarkable as for the DD-ROM, we want to highlight that the accuracy of the GNN solutions are qualitatively much better than the DD-ROM for that viscosity level, and physically more meaningful. This aspect is a major advantage with respect to classical linear ROMs that is probably worth the loss of computational advantage. In perspective, when dealing with nonlinear and more expensive FOM for different equations, the GNN approach will not require any extra computational costs, while FOMs and ROMs model might need special treatments for the nonlinearity that would make their costs increase.

7 Conclusions

We argue that Friedrichs’ systems represent a valuable framework to study and devise reduced order models of many parametric PDEs at the same time: among them the ones studied in this work and others, like mixed elliptic and hyperbolic problems, complex and time-dependent FS and also nonlinear PDEs whose linearization results in FS, e.g. the Euler equations. The advantages include the availability of *a posteriori* error estimators and the easy to preserve mathematical properties of positivity and symmetry from the full-order formulations to the reduced-order ones. We underlined in section 4.2 how optimally stable reduced-order models can be obtained from the ultraweak formulation. A more efficient numerical solver for Friedrichs’ systems is the hybridized discontinuous Galerkin method [17]. These are possible future directions of research.

Working with discontinuous Galerkin discretizations is not only crucial from the possibly mixed elliptic and hyperbolic nature of Friedrichs’ systems, but also to design domain decomposable reduced-order models with a minimum effort: in fact, penalties at the subdomains interfaces are inherited directly from the full-order models. We demonstrated with numerical experiments the limits and the ranges of application of domain decomposable ROMs: generally, with respect to single domain ROMs, there are benefits only when the model under study is truly decomposable, that is when the parameters affect independently different subdomains and the respective solutions are poorly correlated for unseen parametric instances. The results we showed in our academic benchmarks were obtained with the aim to tackle more complex multi-physics models like fluid-structure interaction systems. A typical application of DD-ROMs for FS is represented by parametric PDEs with a mixed elliptic and hyperbolic nature and possibly solution manifolds more and less linearly approximable respectively. The repartitioning strategies we developed are suited to adapt the reduced local dimension of the linear approximants, especially when the parameters influence only a limited region like in test case ADR 5.3.3. The implementation of *ad hoc* physics inspired indicators can be a future direction of research.

The Friedrichs’ systems formulation itself does not solve the problems caused by a slow decaying Kolmogorov n-width. DD-ROMs can help in this regard, isolating regions with a slow Kolmogorov n-width for which nonlinear approximants can be employed and regions with a fast decaying Kolmogorov n-width for which classical linear projection-based ROMs provide efficient and reliable predictions. Related to this subject and motivated also by the heavy computational resources that graph neural networks require when employed for model order reduction, we introduced a new paradigm for surrogate modeling: the inference with GNNs of vanishing viscosity solutions from a succession of higher viscosity projection-based ROMs. The approach is, of course, general and can be applied to PDEs that are not FS. The crucial hypotheses underneath this approach is the approximability with linear spaces of the solution manifolds corresponding to higher viscosity levels. We showed that the additional computational costs are not too large in our test case in section 6. Possible directions of research include more complex problems and different regularization or filtering choices, other than additional viscous terms.

A Transformation into dissipative system

In some cases, the term $\mathcal{A}^0 = 0$ or property (2b) is not satisfied, but there is a way to recover the previous framework. We want to recover a dissipative [46] or accretive system [52]. For example the linearized Euler equations in entropy variables [78] have $\mathcal{A}^0 = 0$.

The condition of uniform positive definiteness

$$\exists \mu_0, \quad \mathcal{A}^0 + (\mathcal{A}^0)^t - \mathcal{X} \geq 2\mu_0 \mathbb{I}_m \text{ a.e. in } \Omega, \quad (100)$$

is still valid if there exist $\xi \in \mathbb{R}^d$, $\|\xi\| = 1$ and $\beta \in \mathbb{R}$, $\beta > 0$ such that after the transformation

$$v(x) = e^{-\beta \xi \cdot x} z(x), \quad (101)$$

the resulting system

$$\sum_i A^i \partial_i v(x) + \beta \sum_{i=1}^d \xi_i A^i v(x) = e^{-\beta(\xi \cdot x)} f, \quad (102)$$

satisfies, with the newly found $A^0 = \beta \sum_{i=1}^d \xi_i A^i$,

$$\exists \mu_0, \quad A^0 + (A^0)^t - \mathcal{X} = 2\beta \sum_{i=1}^d \xi_i A^i - \mathcal{X} \geq 2\mu_0 \mathbb{I}_m \text{ a.e. in } \Omega. \quad (103)$$

In some cases, such ξ and β exist, for example if the symmetric matrix $\sum_{i=1}^d \xi_i A^i$ has at least one positive eigenvalue for some ξ for almost every $x \in \Omega$, then taking β sufficiently large is enough to satisfy the condition. It is also sufficient that $\sum_{i=1}^d \xi_i(x) A^i(x)$ has at least a positive eigenvalue for almost every $x \in \Omega$ where $\xi = \xi(x)$, see [52, Example 28].

Remark 2. A more general transformation is

$$v(x) = w(x)z(x), \quad (104)$$

so that the positive definiteness condition becomes

$$\exists \mu_0, \quad \mathcal{A}^0 + (\mathcal{A}^0)^t - \mathcal{X} = 2 \sum_{i=1}^d \partial_i(-\log w) \mathcal{A}^i - \mathcal{X} \geq 2\mu_0 \mathbb{I}_m \text{ a.e. in } \Omega. \quad (105)$$

B Constructive method to define boundary operators

We report a procedure to define a boundary operator $M \in \mathcal{L}(V, V')$ starting from some specified boundary conditions. We exploit Theorem 4.3, Lemma 4.4 and Corollary 4.1 from [31]. It can be seen that the most common Dirichlet, Neumann and Robin boundary conditions can be found for some FS [28, 29, 24], following this procedure.

Lemma 1 (Theorem 4.3, Lemma 4.4 and Corollary 4.1 from [31]). *Let us assume that (V_0, V_0^*) satisfy (10) and that $V_0 + V_0^* \subset V$ is closed. We denote with $P : V \rightarrow V_0$ and $Q : V \rightarrow V_0^*$ the projectors onto the subspaces $V_0 \subset V$ and $V_0^* \subset V$ of the Hilbert space V , respectively. Then, the boundary operator $\mathcal{M} \in \mathcal{L}(V, V')$ defined as*

$$\begin{aligned} \langle \mathcal{M}u, v \rangle_{V', V} &= \langle \mathcal{D}Pu, Pv \rangle_{V', V} - \langle \mathcal{D}Qu, Qv \rangle_{V', V} + \\ &\quad \langle \mathcal{D}(P+Q-PQ)u, v \rangle_{V', V} - \langle \mathcal{D}u, (P+Q-PQ)v \rangle_{V', V} \end{aligned} \quad (106)$$

is admissible and satisfies $V_0 = \ker(\mathcal{D} - \mathcal{M})$ and $V_0^* = \ker(\mathcal{D} + \mathcal{M}^*)$. In particular,

1. If $V = V_0 + V_0^*$, then \mathcal{M} is self-adjoint and

$$\langle \mathcal{M}u, v \rangle_{V', V} = \langle \mathcal{D}Pu, Pv \rangle_{V', V} - \langle \mathcal{D}Qu, Qv \rangle_{V', V}. \quad (107)$$

2. If $V_0 = V_0^*$, then \mathcal{M} is skew-symmetric and

$$\langle \mathcal{M}u, v \rangle_{V', V} = \langle \mathcal{D}Pu, v \rangle_{V', V} - \langle \mathcal{D}Pv, u \rangle_{V', V}. \quad (108)$$

We remark that, for fixed (V_0, V_0^*) , admissible boundary operators $\mathcal{M} \in \mathcal{L}(V, V')$ that satisfy $V_0 = \ker(\mathcal{D} - \mathcal{M})$ and $V_0^* = \ker(\mathcal{D} + \mathcal{M}^*)$ are not unique. The boundary operator defined in Lemma 1 is just a possible explicit definition, in general.

As an exercise, we show how to find the definition of the operator \mathcal{M} for our linear compressible elasticity FS, from Section 2.1.2. We want to impose the boundary conditions $\mathbf{u}|_{\Gamma_D} = 0$ and $(\boldsymbol{\sigma} \cdot \mathbf{n})|_{\Gamma_N} = 0$, so,

$$V_0 = V_0^* = \{(\mathbf{u}, \boldsymbol{\sigma}) \in V \mid \mathbf{u}|_{\Gamma_D} = 0, (\boldsymbol{\sigma} \cdot \mathbf{n})|_{\Gamma_N} = 0\} = H_{\boldsymbol{\sigma}, \Gamma_N} \times [H_{\Gamma_D}^1(\Omega)]^d, \quad (109)$$

since we defined $V = H_{\boldsymbol{\sigma}} \times [H^1(\Omega)]^d$, with $H_{\boldsymbol{\sigma}} = \{\boldsymbol{\sigma} \in [L^2(\Omega)]^{d \times d} \mid \nabla \cdot (\boldsymbol{\sigma} + \boldsymbol{\sigma}^t) \in [L^2(\Omega)]^d\}$, the traces $\gamma_D : [H^1(\Omega)]^d \rightarrow [H^{\frac{1}{2}}(\Gamma_D)]^d$ and $\gamma_N : H_{\boldsymbol{\sigma}}^d \rightarrow [H^{-\frac{1}{2}}(\Gamma_N)]^d$ on Γ_D and Γ_N are well-defined. In particular,

$$H_{\boldsymbol{\sigma}, \Gamma_N} = \{\boldsymbol{\sigma} \in H_{\boldsymbol{\sigma}} \mid \gamma_N(\boldsymbol{\sigma}) = (\boldsymbol{\sigma} \cdot \mathbf{n})|_{\Gamma_N} = 0\}, \quad [H_{\Gamma_D}^1(\Omega)]^d = \{\mathbf{u} \in [H^1(\Omega)]^d \mid \gamma_D(\mathbf{u}) = \mathbf{u}|_{\Gamma_D} = 0\}. \quad (110)$$

Moreover, (V_0, V_0^*) satisfy the properties of cone formalism (10). Thus, we can use the definition (108) of Lemma 1:

$$\begin{aligned} \langle \mathcal{M}(\boldsymbol{\sigma}, \mathbf{u}), (\boldsymbol{\tau}, \mathbf{v}) \rangle_{V', V} &= \langle \mathcal{D}P(\boldsymbol{\sigma}, \mathbf{u}), (\boldsymbol{\tau}, \mathbf{v}) \rangle_{V', V} - \langle \mathcal{D}P(\boldsymbol{\tau}, \mathbf{v}), (\boldsymbol{\sigma}, \mathbf{u}) \rangle_{V', V} \\ &= -\langle \frac{1}{2}(\boldsymbol{\sigma} + \boldsymbol{\sigma}^t) \cdot \mathbf{n}, \mathbf{v} \rangle_{-\frac{1}{2}, \frac{1}{2}, \Gamma_D} + \langle \frac{1}{2}(\boldsymbol{\tau} + \boldsymbol{\tau}^t) \cdot \mathbf{n}, \mathbf{u} \rangle_{-\frac{1}{2}, \frac{1}{2}, \Gamma_D} + \\ &\quad \langle \frac{1}{2}(\boldsymbol{\sigma} + \boldsymbol{\sigma}^t) \cdot \mathbf{n}, \mathbf{v} \rangle_{-\frac{1}{2}, \frac{1}{2}, \Gamma_N} - \langle \frac{1}{2}(\boldsymbol{\tau} + \boldsymbol{\tau}^t) \cdot \mathbf{n}, \mathbf{u} \rangle_{-\frac{1}{2}, \frac{1}{2}, \Gamma_N}, \end{aligned} \quad (111)$$

where $P : V = H_{\boldsymbol{\sigma}} \times [H^1(\Omega)]^d \rightarrow V_0 = H_{\boldsymbol{\sigma}, \Gamma_N} \times [H_{\Gamma_D}^1(\Omega)]^d$ is the projector into the subspace V_0 of the Hilbert graph space V with scalar product:

$$\langle (\boldsymbol{\sigma}, \mathbf{u}), (\boldsymbol{\tau}, \mathbf{v}) \rangle_V = (\mathbf{u}, \mathbf{v})_{[L^d(\Omega)]^d} + (\boldsymbol{\sigma}, \boldsymbol{\tau})_{[L^2(\Omega)]^{d \times d}} + (A(\boldsymbol{\sigma}, \mathbf{u}), A(\boldsymbol{\tau}, \mathbf{v})). \quad (112)$$

C ROM convergence studies

In this section, we validate the DD-ROM implementation, checking the convergence towards the FOM solutions with respect to the dimension of the reduced space. Uniform local reduced dimensions are employed $\{r_{\Omega_i}\}_{i=1}^K$ and $\{r_{\Omega_i}\}_{i=1}^k$. For each convergence study 20 uniformly independent samples are used as training dataset and 50 uniformly independent samples as test dataset.

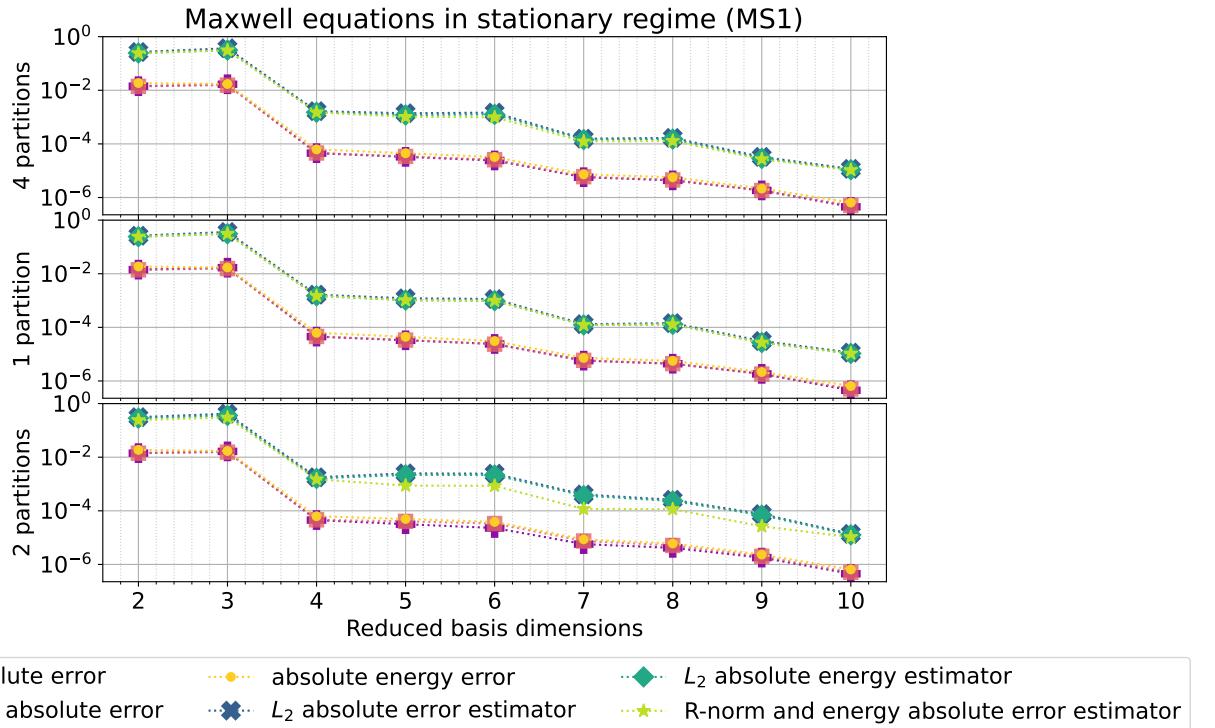


Figure 23: **MS1**. The convergence of DD-ROMS with uniform local reduced dimensions $\{r_{\Omega_i}\}_{i=1}^K$ and $\{r_{\Omega_i}\}_{i=1}^k$ is assessed. The uniform value of the local reduced dimensions is reported in the abscissae. For this test case an improvement of the accuracy with respect to the single domain reduced basis is not observed.

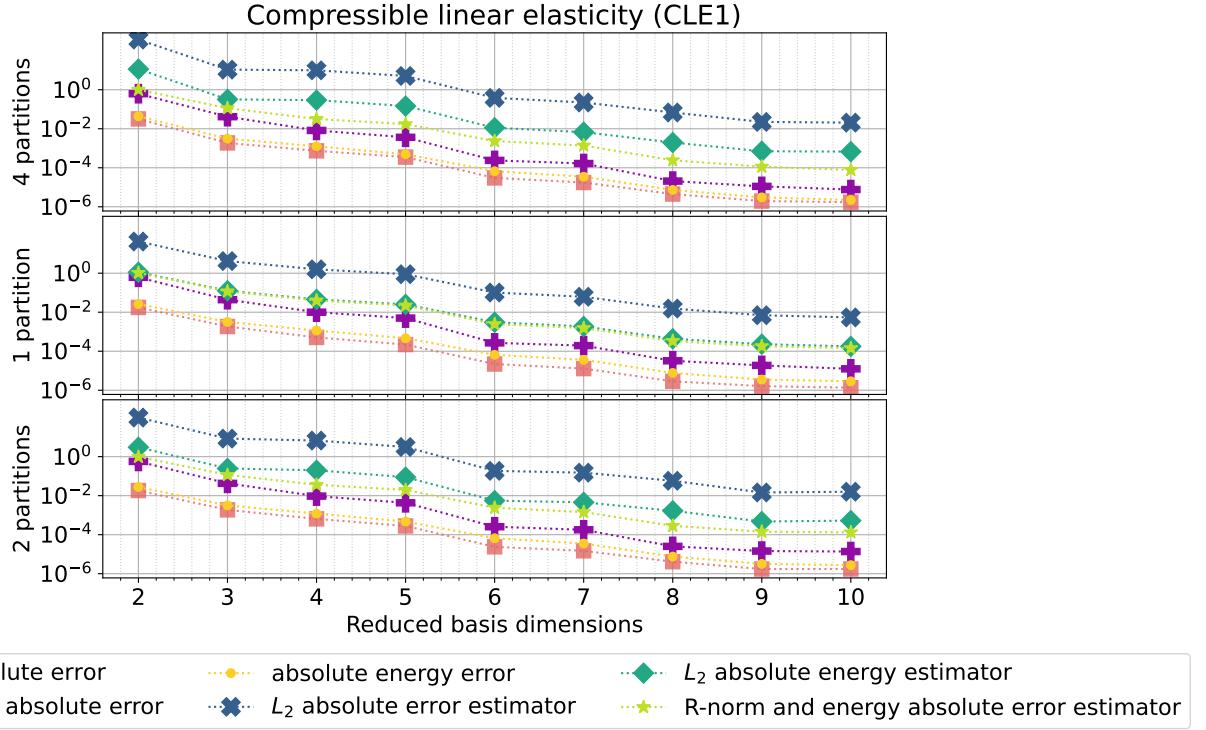


Figure 24: **CLE1**. The convergence of DD-ROMS with uniform local reduced dimensions $\{r_{\Omega_i}\}_{i=1}^K$ and $\{r_{\Omega_i}\}_{i=1}^k$ is assessed. The uniform value of the local reduced dimensions is reported in the abscissae. For this test case an improvement of the accuracy with respect to the single domain reduced basis is not observed.

In Figure 23, we show the L^2 -error, the R -error and the energy error decay and their respective error estimators for the Maxwell equations test case **MS1**, section 5.3.1, with constant parameters μ and σ on the whole domain. We clearly see an exponential behavior in the error as we add basis functions. On the other hand, we do not observe strong differences between the ROM, DD-ROM with repartition and DD-ROM with `deal.II` subdomains, for this simple test case. Similar results can be observed in Figure 24, where the same analysis is applied for the compressible linear elasticity test **CLE1** from section 5.3.2.

From these results, it should be clear that the employment of local reduced basis is not always useful to increase the accuracy of the predictions. Nonetheless, it may be used to locally reduce the dimension of the linear approximants. Possible benefits include the adaptation of the computational resources (higher dimensional reduced basis are chosen only where it is necessary) and the possibility to speedup parametric studies and non-intrusive surrogate modelling thanks to the further reduced local dimensions [85, 86].

Typical cases where DD-ROMs are effective to increase the accuracy of the predictions are truly decomposable systems where the parameters affect independently different regions of the computational domain, as in test case **MS2** in section 5.3.1.

Acknowledgements

This work was partially funded by European Union Funding for Research and Innovation — Horizon 2020 Program — in the framework of European Research Council Executive Agency: H2020 ERC CoG 2015 AROMA-CFD project 681447 “Advanced Reduced Order Methods with Applications in Computational Fluid Dynamics” P.I. Professor Gianluigi Rozza. We also acknowledge the PRIN 2017 “Numerical Analysis for Full and Reduced Order Methods for the efficient and accurate solution of complex systems governed by Partial Differential Equations” (NA-FROM-PDEs). Davide Torlo has been funded by a SISSA Mathematical fellowship within Italian Excellence Departments initiative by Ministry of University and Research.

References

- [1] P. R. Amestoy, A. Guermouche, J.-Y. L'Excellent, and S. Pralet. Hybrid scheduling for the parallel solution of linear systems. *Parallel Computing*, 32(2):136–156, 2006.

- [2] D. Amsallem and C. Farhat. Interpolation method for adapting reduced-order models and application to aeroelasticity. *Aiaa Journal - AIAA J*, 46:1803–1813, 07 2008.
- [3] N. Antonic and K. Burazin. On equivalent descriptions of boundary conditions for Friedrichs systems. *Math. Montisnigri*, 22(23):5–13, 2009.
- [4] N. Antonić, K. Burazin, I. Crnjac, and M. Erceg. Complex Friedrichs systems and applications. *Journal of Mathematical Physics*, 58(10):101508, 2017.
- [5] F. Arbes, C. Greif, and K. Urban. The Kolmogorov N-width for linear transport: Exact representation and the influence of the data. *arXiv preprint arXiv:2305.00066*, 2023.
- [6] D. Arndt, W. Bangerth, B. Blais, M. Fehling, R. Gassmöller, T. Heister, L. Heltai, U. Köcher, M. Kronbichler, M. Maier, P. Munch, J.-P. Pelteret, S. Proell, K. Simon, B. Turcksin, D. Wells, and J. Zhang. The `deal.II` library, version 9.3. *Journal of Numerical Mathematics*, 29(3):171–186, 2021.
- [7] J. Baiges, R. Codina, and S. Idelsohn. A domain decomposition strategy for reduced order models. Application to the incompressible Navier–Stokes equations. *Computer Methods in Applied Mechanics and Engineering*, 267:23–42, 2013.
- [8] S. Balay, S. Abhyankar, M. F. Adams, S. Benson, J. Brown, P. Brune, K. Buschelman, E. M. Constantinescu, L. Dalcin, A. Dener, V. Eijkhout, J. Faibussowitsch, W. D. Gropp, V. Hapla, T. Isaac, P. Jolivet, D. Karpeev, D. Kaushik, M. G. Knepley, F. Kong, S. Kruger, D. A. May, L. C. McInnes, R. T. Mills, L. Mitchell, T. Munson, J. E. Roman, K. Rupp, P. Sanan, J. Sarich, B. F. Smith, S. Zampini, H. Zhang, H. Zhang, and J. Zhang. PETSc Web page. <https://petsc.org/>, 2023.
- [9] W. Bangerth, C. Burstedde, T. Heister, and M. Kronbichler. Algorithms and data structures for massively parallel generic adaptive finite element codes. *ACM Trans. Math. Softw.*, 38:14/1–28, 2011.
- [10] C. Beattie, S. Gugercin, and V. Mehrmann. Structure-preserving interpolatory model reduction for port-Hamiltonian differential-algebraic systems. In *Realization and Model Reduction of Dynamical Systems: A Festschrift in Honor of the 70th Birthday of Thanos Antoulas*, pages 235–254. Springer, 2022.
- [11] E. Beurer, M. Feuerle, N. Reich, and K. Urban. An ultraweak variational method for parameterized linear differential-algebraic equations. *Frontiers in Applied Mathematics and Statistics*, 8:910786, 2022.
- [12] T. Bui-Thanh, L. Demkowicz, and O. Ghattas. A unified discontinuous Petrov–Galerkin method and its analysis for Friedrichs’ systems. *SIAM Journal on Numerical Analysis*, 51(4):1933–1958, 2013.
- [13] T. Bui-Thanh, L. Demkowicz, and O. Ghattas. Constructively well-posed approximation methods with unity inf–sup and continuity constants for partial differential equations. *Mathematics of Computation*, 82(284):1923–1952, 2013.
- [14] C. Burstedde, L. C. Wilcox, and O. Ghattas. `p4est`: Scalable algorithms for parallel adaptive mesh refinement on forests of octrees. *SIAM Journal on Scientific Computing*, 33(3):1103–1133, 2011.
- [15] N. Cagniart, Y. Maday, and B. Stamm. *Model Order Reduction for Problems with Large Convection Effects*, pages 131–150. Springer International Publishing, Cham, 2019.
- [16] K. Carlberg, C. Bou-Mosleh, and C. Farhat. Efficient non-linear model reduction via a least-squares Petrov–Galerkin projection and compressive tensor approximations. *International Journal for Numerical Methods in Engineering*, 86(2):155–181, 2011.
- [17] J.-U. Chen, S. Kang, T. Bui-Thanh, and J. N. Shadid. Unified hp -HDG Frameworks for Friedrichs’ PDE systems. *arXiv preprint arXiv:2304.03690*, 2023.
- [18] A. Cohen and R. DeVore. Approximation of high-dimensional parametric PDEs. *Acta Numerica*, 24:1–159, 2015.
- [19] A. Cohen and R. DeVore. Kolmogorov widths under holomorphic mappings. *IMA Journal of Numerical Analysis*, 36(1):1–12, 2016.
- [20] R. Crisovan, D. Torlo, R. Abgrall, and S. Tokareva. Model order reduction for parametrized nonlinear hyperbolic problems as an application to uncertainty quantification. *Journal of Computational and Applied Mathematics*, 348:466–489, 2019.
- [21] L. Dalcin and Y.-L. L. Fang. mpi4py: Status update after 12 years of development. *Computing in Science and Engineering*, 23(4):47–54, 2021.

- [22] L. D. Dalcin, R. R. Paz, P. A. Kler, and A. Cosimo. Parallel distributed computing using Python. *Advances in Water Resources*, 34(9):1124 – 1139, 2011. New Computational Methods and Software Tools.
- [23] M. B. del Pino, T. C. Rebollo, and M. G. Márquez. A boundary-oriented reduced Schwarz domain decomposition technique for parametric advection-diffusion problems. *arXiv preprint arXiv:2305.19199*, 2023.
- [24] D. A. Di Pietro and A. Ern. *Mathematical aspects of discontinuous Galerkin methods*, volume 69. Springer Science & Business Media, 2011.
- [25] R. J. DiPerna. Convergence of approximate solutions to conservation laws. In *Transonic, Shock, and Multidimensional Flows*, pages 313–328. Elsevier, 1982.
- [26] R. J. DiPerna. Convergence of the viscosity method for isentropic gas dynamics. *Communications in mathematical physics*, 91(1):1–30, 1983.
- [27] J. Eftang, D. Huynh, D. Knezevic, E. Ronquist, and A. Patera. Adaptive port reduction in static condensation. *IFAC Proceedings Volumes*, 45(2):695–699, 2012.
- [28] A. Ern and J.-L. Guermond. Discontinuous Galerkin methods for Friedrichs' systems. I. General theory. *SIAM journal on numerical analysis*, 44(2):753–778, 2006.
- [29] A. Ern and J.-L. Guermond. Discontinuous Galerkin Methods for Friedrichs' Systems. Part II. Second-order Elliptic PDEs. *SIAM Journal on Numerical Analysis*, 44(6):2363–2388, Jan. 2006.
- [30] A. Ern and J.-L. Guermond. Discontinuous Galerkin methods for Friedrichs' systems. Part III. Multifield theories with partial coercivity. *SIAM journal on numerical analysis*, 46(2):776–804, 2008.
- [31] A. Ern, J.-L. Guermond, and G. Caplain. An intrinsic criterion for the bijectivity of Hilbert operators related to Friedrichs' systems. *Communications in partial differential equations*, 32(2):317–341, 2007.
- [32] M. Fey and J. E. Lenssen. Fast graph representation learning with PyTorch Geometric. In *ICLR Workshop on Representation Learning on Graphs and Manifolds*, 2019.
- [33] N. R. Franco, A. Manzoni, and P. Zunino. Learning operators with mesh-informed neural networks. *arXiv preprint arXiv:2203.11648*, 2022.
- [34] K. O. Friedrichs. Symmetric positive linear differential equations. *Communications on Pure and Applied Mathematics*, 11(3):333–418, 1958.
- [35] K. Fukami, K. Fukagata, and K. Taira. Super-resolution reconstruction of turbulent flows with machine learning. *Journal of Fluid Mechanics*, 870:106–120, 2019.
- [36] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl. Neural message passing for quantum chemistry. In *International conference on machine learning*, pages 1263–1272. PMLR, 2017.
- [37] E. Godlewski and P.-A. Raviart. *Hyperbolic systems of conservation laws*. Number 3-4. Ellipses, 1991.
- [38] J. Goodman and Z. Xin. Viscous limits for piecewise smooth solutions to systems of conservation laws. *Archive for rational mechanics and analysis*, 121:235–265, 1992.
- [39] S. Hain and K. Urban. An ultra-weak space-time variational formulation for the Schrödinger equation. *arXiv preprint arXiv:2212.14398*, 2022.
- [40] W. Hamilton, Z. Ying, and J. Leskovec. Inductive representation learning on large graphs. *Advances in neural information processing systems*, 30, 2017.
- [41] C. R. Harris, K. J. Millman, S. J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N. J. Smith, R. Kern, M. Picus, S. Hoyer, M. H. van Kerkwijk, M. Brett, A. Haldane, J. F. del Río, M. Wiebe, P. Peterson, P. Gérard-Marchant, K. Sheppard, T. Reddy, W. Weckesser, H. Abbasi, C. Gohlke, and T. E. Oliphant. Array programming with NumPy. *Nature*, 585(7825):357–362, Sept. 2020.
- [42] J. Henning, D. Palitta, V. Simoncini, and K. Urban. An ultraweak space-time variational formulation for the wave equation: Analysis and efficient numerical solution. *ESAIM: Mathematical Modelling and Numerical Analysis*, 56(4):1173–1198, 2022.

- [43] J. S. Hesthaven, C. Pagliantini, and N. Ripamonti. Structure-preserving model order reduction of Hamiltonian systems. *arXiv preprint arXiv:2109.12367*, 2021.
- [44] J. S. Hesthaven, G. Rozza, B. Stamm, et al. *Certified reduced basis methods for parametrized partial differential equations*, volume 590. Springer, 2016.
- [45] C. Hoang, Y. Choi, and K. Carlberg. Domain-decomposition least-squares Petrov–Galerkin (DD-LSPG) nonlinear model reduction. *Computer methods in applied mechanics and engineering*, 384:113997, 2021.
- [46] P. Houston, J. A. Mackenzie, E. Süli, and G. Warnecke. A posteriori error analysis for numerical approximations of Friedrichs systems. *Numerische Mathematik*, 82(3):433–470, 1999.
- [47] T. J. Hughes, M. Mallet, and M. Akira. A new finite element formulation for computational fluid dynamics: II. Beyond SUPG. *Computer methods in applied mechanics and engineering*, 54(3):341–355, 1986.
- [48] D. B. P. Huynh, D. J. Knezevic, and A. T. Patera. A static condensation reduced basis element method: approximation and a posteriori error estimation. *ESAIM: Mathematical Modelling and Numerical Analysis*, 47(1):213–251, 2013.
- [49] A. Iollo and D. Lombardi. Advection modes by optimal mass transfer. *Phys. Rev. E*, 89:022923, Feb 2014.
- [50] A. Iollo, G. Sambataro, and T. Taddei. A one-shot overlapping Schwarz method for component-based model reduction: application to nonlinear elasticity. *Computer Methods in Applied Mechanics and Engineering*, 404:115786, 2023.
- [51] 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.
- [52] M. Jensen. *Discontinuous Galerkin methods for Friedrichs systems with irregular solutions*. PhD thesis, Citeseer, 2004.
- [53] I. T. Jolliffe. *Principal component analysis for special types of data*. Springer, 2002.
- [54] G. Karypis and V. Kumar. *METIS: A Software Package for Partitioning Unstructured Graphs, Partitioning Meshes, and Computing Fill-Reducing Orderings of Sparse Matrices*, September 1998.
- [55] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- [56] D. Kochkov, J. A. Smith, A. Alieva, Q. Wang, M. P. Brenner, and S. Hoyer. Machine learning–accelerated computational fluid dynamics. *Proceedings of the National Academy of Sciences*, 118(21):e2101784118, 2021.
- [57] S. N. Kružkov. First order quasilinear equations in several independent variables. *Mathematics of the USSR-Sbornik*, 10(2):217, 1970.
- [58] K. Kunisch and S. Volkwein. Galerkin proper orthogonal decomposition methods for parabolic problems. *Numerische mathematik*, 90(1):117–148, 2001.
- [59] 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.
- [60] D. J. Lucia, P. I. King, and P. S. Beran. Domain decomposition for reduced-order modeling of a flow with moving shocks. *AIAA journal*, 40(11):2360–2362, 2002.
- [61] Y. Maday and E. M. Rønquist. A reduced-basis element method. *Journal of Scientific Computing*, 17:447–459, 2002.
- [62] Y. Maday and E. M. Ronquist. The reduced basis element method: application to a thermal fin problem. *SIAM Journal on Scientific Computing*, 26(1):240–258, 2004.
- [63] Y. Maday and E. Tadmor. Analysis of the spectral vanishing viscosity method for periodic conservation laws. *SIAM Journal on Numerical Analysis*, 26(4):854–870, 1989.
- [64] O. Oleinik. Discontinuous solutions of nonlinear differential equations. *Amer. Math. Soc. Transl.*, 26(2):95–172, 1963.
- [65] E. J. Parish and F. Rizzi. On the impact of dimensionally-consistent and physics-based inner products for POD-Galerkin and least-squares model reduction of compressible flows. *arXiv preprint arXiv:2203.16492*, 2022.
- [66] 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.

- [67] F. Pichi, B. Moya, and J. S. Hesthaven. A graph convolutional autoencoder approach to model order reduction for parametrized PDEs. *arXiv preprint arXiv:2305.08573*, 2023.
- [68] C. Prud'Homme, D. V. Rovas, K. Veroy, L. Machiels, Y. Maday, A. T. Patera, and G. Turinici. Reliable real-time solution of parametrized partial differential equations: Reduced-basis output bound methods. *J. Fluids Eng.*, 124(1):70–80, 2002.
- [69] C. Prud'Homme, D. V. Rovas, K. Veroy, and A. T. Patera. A mathematical and computational framework for reliable real-time solution of parametrized partial differential equations. *ESAIM: Mathematical Modelling and Numerical Analysis*, 36(5):747–771, 2002.
- [70] I. Prusak, M. Nonino, D. Torlo, F. Ballarin, and G. Rozza. An optimisation-based domain-decomposition reduced order model for the incompressible Navier-Stokes equations. *arXiv preprint arXiv:2211.14528*, 2022.
- [71] I. Prusak, D. Torlo, M. Nonino, and G. Rozza. An optimisation-based domain-decomposition reduced order model for parameter-dependent non-stationary fluid dynamics problems. *arXiv preprint arXiv:2308.01733*, 2023.
- [72] J. Rauch. Symmetric positive systems with boundary characteristic of constant multiplicity. *Transactions of the American Mathematical Society*, 291(1):167–187, 1985.
- [73] J. Rauch. Boundary value problems with nonuniformal characteristic boundary. *Journal de mathématiques pures et appliquées*, 73(4):347–353, 1994.
- [74] D. Rim and K. T. Mandli. Displacement interpolation using monotone rearrangement. *SIAM/ASA Journal on Uncertainty Quantification*, 6(4):1503–1531, 2018.
- [75] F. Romor, M. Tezzele, and G. Rozza. A local approach to parameter space reduction for regression and classification tasks. *arXiv preprint arXiv:2107.10867*, 2021.
- [76] G. Rozza, G. Stabile, and F. Ballarin. *Advanced Reduced Order Methods and Applications in Computational Fluid Dynamics*. SIAM, 2022.
- [77] M. Simonovsky and N. Komodakis. Dynamic edge-conditioned filters in convolutional neural networks on graphs. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3693–3702, 2017.
- [78] T. Sonar and E. Süli. A dual graph-norm refinement indicator for finite volume approximations of the Euler equations. *Numerische Mathematik*, 78(4):619–658, 1998.
- [79] T. Taddei. A registration method for model order reduction: Data compression and geometry reduction. *SIAM Journal on Scientific Computing*, 42(2):A997–A1027, 2020.
- [80] J. Tencer and K. Potter. A tailored convolutional neural network for nonlinear manifold learning of computational physics data using unstructured spatial discretizations. *SIAM Journal on Scientific Computing*, 43(4):A2581–A2613, 2021.
- [81] D. Torlo. Model Reduction for Advection Dominated Hyperbolic Problems in an ALE Framework: Offline and Online Phases. *arXiv e-prints*, page arXiv:2003.13735, Mar. 2020.
- [82] D. Torlo and M. Ricchiuto. Model order reduction strategies for weakly dispersive waves. *Mathematics and Computers in Simulation*, 205:997–1028, 2023.
- [83] A. Van Der Schaft, D. Jeltsema, et al. Port-Hamiltonian systems theory: An introductory overview. *Foundations and Trends® in Systems and Control*, 1(2-3):173–378, 2014.
- [84] D. Wells, Z. Wang, X. Xie, and T. Iliescu. An evolve-then-filter regularized reduced order model for convection-dominated flows. *International Journal for Numerical Methods in Fluids*, 84(10):598–615, 2017.
- [85] D. Xiao, F. Fang, C. E. Heaney, I. Navon, and C. Pain. A domain decomposition method for the non-intrusive reduced order modelling of fluid flow. *Computer Methods in Applied Mechanics and Engineering*, 354:307–330, 2019.
- [86] D. Xiao, C. Heaney, F. Fang, L. Mottet, R. Hu, D. Bistrian, E. Aristodemou, I. Navon, and C. Pain. A domain decomposition non-intrusive reduced order model for turbulent flows. *Computers & Fluids*, 182:15–27, 2019.
- [87] X. Xie, D. Wells, Z. Wang, and T. Iliescu. Numerical analysis of the Leray reduced order model. *Journal of Computational and Applied Mathematics*, 328:12–29, 2018.