Archives of Computational Methods in Engineering
https://doi.org/10.1007/s11831-026-10690-9

REVIEW

A Structured Review of Reduced Order Modeling for Domain Decomposition Problems: State of the Art and Perspectives

Shenhui Ruan[1] · **Andreas G. Class**[1] · **Gianluigi Rozza**[2]

Received: 13 January 2026 / Revised: 27 May 2026 / Accepted: 7 June 2026

Abstract

Reduced order models (ROMs) have been regarded as an efficient alternative to conventional high-fidelity methods for accelerating the design and optimization processes in engineering applications. Many industrial facilities feature repeating geometrical patterns or contain subregions governed by distinct physical phenomena, making them well-suited to Domain Decomposition (DD) techniques. The integration of a ROM and DD is promising to further reduce computational costs by constructing local ROMs and assembling them into global solutions. Due to the complexity and necessity of coupling ROMs, many approaches have been proposed in recent years. This review provides a concise overview of existing methodologies combining ROMs and DD. We categorize existing methods into intrusive (projection-based) and non-intrusive (data-driven) frameworks. Various strategies for generating local reduced bases and coupling them across subdomains are illustrated. Particular emphasis is placed on intrusive techniques, including equations, numerical algorithms, and practical implementations. The non-intrusive framework is also discussed, highlighting its general procedures, basic formulations, and underlying principles. Finally, we summarize the state of the literature, identify open challenges, and present perspectives on future implementations from an engineering viewpoint.

1 Introduction and Background

Numerical analysis, and especially high-fidelity three-dimensional simulations, have recently become practical tools for the design and optimization of various industrial applications, such as power engineering [1], [2], and energy storage systems [3]. In comparison with experiments, which are expensive and time-consuming for large-scale facilities, simulations are economically efficient. They are practical for improving efficiency and estimating the safety threshold in many applications. Over the past several decades, many numerical schemes have been developed, such as the finite element, finite volume, spectral element, and lattice Boltzmann methods [4–7], to fulfill specific demands of analysis in various fields.

Nevertheless, even with the fast development of hardware, multi-query and multi-scale simulation of large-scale systems is still challenging. Therefore, even today, a large range of parametric studies, uncertainty, and sensitivity analyses are limited by computational resources. In spite of this, the demand for fast modeling and real-time control is increasing in almost every industry. To overcome this limitation, *ROMs* have been proposed. They can accelerate simulations by several orders of magnitude. Thus, they have been regarded as efficient techniques to obtain high-resolution approximations for complex physics [8–11]. Note that in the context of the ROM, the corresponding high-fidelity model is called the *Full Order Model* (FOM).

The fundamental philosophy of ROMs is that high-dimensional solutions over a range of parameters can be well approximated using a few sampled solutions. The implementation of ROMs consists of two steps: (i) a set of expensive *offline* calculations is carried out for optimally selected conditions; (ii) real-time *online* calculations are then performed for numerous cases [12].

Shenhui Ruan
shenhui.ruan@kit.edu

Andreas G. Class
andreas.class@kit.edu

Gianluigi Rozza
grozza@sissa.it

[1] Institute for Thermal Energy Technology and Safety, Karlsruhe Institute of Technology, Kaiserstraße 12, 76131 Karlsruhe, Germany

[2] Mathematics Area, mathLab, International School for Advanced Studies, via Bonomea 265, 34136 Trieste, Italy

During the offline phase, we collect FOM solutions (called *snapshots*) for the sampled parameters. These snapshots are assembled column-by-column into a snapshot matrix. Since the FOM usually has a large number of *Degrees of Freedom* (DoFs), the matrix becomes high-dimensional (numerous rows). Therefore, dimensionality reduction is applied to retain only a few dominant features while preserving the essential dynamics of the flow. The reduction of datasets can be achieved utilizing various techniques, including traditional methods like *Proper Orthogonal Decomposition* (POD) and the iterative *greedy* algorithm [12–14], as well as more recent *neural network*-based techniques [15].

Once the reduced representations are available, ROMs are constructed based on them. Over the past several decades, many ROM construction techniques have been developed, generally classified as (i) intrusive and (ii) non-intrusive methods [9, 16, 17]. The former manipulate the governing *Partial Differential Equations* (PDEs) using *(Petrov-)Galerkin projection*[1] to derive a reduced system. In this case, the dominant physical features are known as basis functions because they can span a *Reduced Basis* (RB) of high-fidelity solutions. The latter are purely data-driven and adopt a set of sampled data to *train* a surrogate model (i.e., a ROM) to *predict* solutions for unknown parameters. In both frameworks, the resulting ROMs are validated against an unseen dataset to evaluate their predictive capability.

Traditionally, “ROM” often denotes intrusive methods, while purely data-driven approaches are commonly called “surrogate models”. In this paper, we adopt the convention that “ROM” refers to intrusive projection-based methods, while for data-driven techniques, we use the term “surrogate model” and “ROM” interchangeably, depending on the context. When necessary, we explicitly refer to data-driven methods as “non-intrusive ROMs” to maintain clarity.

Now, we turn our attention to the treatment of the geometries in ROMs, a fundamental topic of this review. Classic ROMs approximate the entire computational domain, which is known as *Global ROMs*. However, this direct method has certain drawbacks, as indicated in [18, 19]:

- The RB is optimized to capture the most necessary flow dynamics across the entire model. However, representing numerous local phenomena requires a large number of modes, which weakens the advantage of ROMs.[2]
- High computational costs may arise in the offline stage due to the need for generating solutions in huge computational domains and performing dimensional reduction on extensive datasets.
- Increasing POD modes can improve the theoretical approximation accuracy, but stability is not guaranteed without appropriate stabilization techniques, as indicated in [20, 21].

These shortcomings have led the research community to integrate DD techniques into ROMs. To reduce computational complexity, this approach divides the physical domain into subdomains. First, reduced spaces are computed locally. Then, the *local fields* are *glued* together to construct an accurate global approximation. Compared to the global ROM, the local approach (referred to as a local ROM or a DD-ROM hereafter) offers several advantages [22–25]:

- Low-dimensional representations are computed at the local level, leading to a better representation of dominant variations in each subdomain.
- Different parameterizations can be applied locally, enhancing the flexibility.
- Computational costs decrease in high-fidelity simulations, dimensionality reduction of large snapshot datasets, and the training stage of ROMs.
- Different numbers of modes or distinct reduced subspaces can be assigned to each subdomain, reducing overall computational costs in the prediction stage while enhancing robustness and stability.

We emphasize that the local ROM is well-suited for various realistic applications. Firstly, geometries consisting of repeating patterns are widely employed in many practical applications. Examples include fission [26–28] and fusion [29] nuclear reactors, heat exchangers [30], fuel cells [31], thermal storage systems [32], and biological models [33], among others. Due to symmetry and periodicity, physical phenomena within these repeating structures often exhibit similar behavior. Thus, domain decomposition is applicable to the treatment of such geometries.

Moreover, for large-scale simulations, different regions may require varying levels of accuracy. FOMs frequently utilize domain decomposition with varying mesh resolutions across partitions, necessitating corresponding adjustments in ROMs to handle non-conformal mesh interfaces. Consequently, domain decomposition is often applied at both the FOM and ROM levels [34]. Additionally, coupling FOMs with ROMs also relies on spatial partitioning [35, 36].

This technique is particularly advantageous in multi-physics simulations, where different regions are governed by distinct PDEs. A key example is *Fluid-Structure Interaction*

---

[1] There are several variations of the Galerkin projection, including the standard Galerkin and Petrov-Galerkin methods. See the explanation in Sect. 4.1.

[2] This issue is called a slow decay of the *Kolmogorov n-width* for a ROM [12].

(FSI) [37–39], where localized solution methodologies naturally align with partitioned domains.

Given its relevance to our experience, we emphasize that spatial partitioning is widely adopted in energy-related applications, such as the subchannel approach and the coarse grid computational fluid dynamics method. The subchannel analysis [40, 41], commonly used in nuclear engineering, divides the computational domain into macro control volumes, where interactions and small-scale effects are incorporated through empirically derived correlations. This partitioning strategy can be combined with local ROM techniques for improved nuclear engineering simulations [27]. The coarse grid method [42], on the other hand, employs a highly coarse discretization of the spatial domain, capturing sub-grid effects through source terms. These terms are estimated via interpolated data from high-resolution simulations, making the approach effective for modeling flow in tube bundles [43] and wind farms [42].

Although subchannel analysis and the coarse grid method are effective for specific applications, they often rely on empirical correlations and may not capture all the local dynamics of the system. Additionally, their solutions are typically low-resolution, which can limit their predictive capabilities. In contrast, ROMs achieve high-fidelity solutions with significantly reduced computational cost, making them a compelling alternative for large-scale simulations that require both accuracy and efficiency.

The development of ROMs for domain decomposition problems began around the early 2000s and gained significant traction in the past decade. They have gathered significant research interest and show promising potential for scientific and engineering applications. They have been used to model various PDEs, including the Laplace equation [44], the time-dependent heat equation [45], convection-diffusion [46], solid mechanics [47], fluid dynamics, particularly the Navier-Stokes equations [48], and multiphysics problems [49].

Two comprehensive reviews for this topic have been published. Buhr et al. [50] published a thorough review of intrusive localized ROM approaches in 2019. Another review regarding non-intrusive procedures was published in 2021 by Heinlein et al. [51], which discusses a combination of machine learning techniques and domain decomposition.

However, various studies have appeared in recent years. Advanced techniques based on neural networks have been intensively applied to the topic. We observe that a recent in-depth review of the field is lacking. Therefore, this paper provides a structured overview of existing work on local ROMs from an engineering and practical perspective.

Our goal is to categorize both classical and advanced techniques developed over the past decades and present fundamental aspects of their implementation. We first identify three common aspects that apply to all methods: domain decomposition, parameterization, and dimensionality reduction. As these three items are not unique, they are classified and discussed separately. Then, we review both intrusive and non-intrusive frameworks, with each category further subdivided according to the mathematical formulation and/or numerical algorithm. This hierarchical classification is well-suited for a structured comparison of various approaches, highlighting their unique characteristics and applications.

The review is organized as follows. Section 2 presents the overall outline, which classifies existing techniques with regard to domain decomposition, parameterization, snapshot reduction, and coupling algorithms. Section 3.1 presents strategies for domain decomposition. Treatments for parameter space are briefly addressed in Sect. 3.2. Various procedures for generating snapshots and computing local RBs are illustrated in Sect. 3.3. Section 4 introduces projection-based coupling methods, including monolithic and iterative algorithms. The explanation and overview considering pure data-driven approaches are presented in Sect. 5. Finally, Sect. 6 summarizes the review and presents conclusions on the state of the art and challenges in ROM techniques for domain decomposition problems.

2 Classification and Hierarchy

In summary, four general aspects arise and must be considered prior to constructing a local ROM: (i) partition strategies, (ii) parameterization techniques, (iii) local dimensionality reduction procedures, and (iv) acceleration techniques for the offline and online stages. Additionally, various algorithms have been proposed to reconstruct global fields by coupling local approximations. Therefore, to better organize the review, the categorization and hierarchy for the observed references are presented as follows.

2.1 Domain Decomposition Strategies

The domain can be decomposed into *overlapping* or *non-overlapping* parts, which mainly depends on the coupling techniques (see Sect. 3.1.1). Also, the high-resolution mesh can be *conforming* or *non-conforming* at partition interfaces (shown in Sect. 3.1.2). Complex structures can be decomposed into many subdomains of different shapes. However, for geometries with repeating patterns, one can adopt several types of *generic* (or *archetype*) blocks to assemble the whole region. A geometric transformation will map the *reference* subdomains into their final position and shape. More specific explanations are presented in Sect. 3.1.3.

2.2 Parameterization

ROMs are built based on a set of high-fidelity simulations. Each solution is carried out considering some conditions, which we interpret as parameters, sampled from a specific space. More precisely, one should consider how to select values within the specified range. Since this selection is crucial for the quality and performance of both global and local ROMs, Sect. 3.2.1 presents a brief overview of *sampling strategies*. Note that, for the domain decomposition configuration, each subdomain can be parametrized separately, which enhances flexibility for complex domains.

Identifying the parameters is also important. Physical properties and boundary conditions can be interpreted directly, as they are already involved in the governing PDEs. Additionally, the shape of the computational domain can be considered a parameter, which is necessary when exploiting ROMs for tasks such as shape optimization. Nevertheless, generating geometrical samples is not as straightforward as the physical ones. Thus, Sect. 3.2.2 illustrates two widely used techniques for achieving the task, namely, *interpolation-based* methods and a more general approach called *Free Form Deformation* (FFD). We note that the geometric parameterization can be utilized globally and locally for a computational domain. If a model is decomposed into subdomains, it is also possible to deform each one individually.

The dimension of the parameter space is another aspect in this context. For simple conditions, sampling can be performed along every coordinate. However, for a complex scenario with numerous parameters, too many samples result in prohibitively high costs. Hence, Sect. 3.2.3 provides two approaches for dimensionality reduction of the parameter space: *active subspaces* and *generative models* based on *Artificial Intelligence* (AI).

Moreover, for a wide range of the parameter space, a single RB may not be feasible to capture the dominant behaviors across the entire range. Therefore, the parameter space or the offline snapshot dataset can be partitioned into several clusters. Then, a *local RB* is computed for each cluster, as described in Sect. 3.2.4.

2.3 Dimensionality Reduction

After decomposition and sampling, we can begin simulations, collect datasets, and extract dominant features. We note that the snapshots can be collected either globally or locally, which in turn affects the dimensionality reduction. Be aware that the RB is adopted for the following description. However, the illustration also applies to data-driven frameworks.

In the case of the *global snapshots*, the RBs can be computed in three ways: (i) the global RB is computed and then divided into many local parts following the domain decomposition (as presented in Sect. 3.3.1); (ii) the global snapshot matrix is split into several submatrices corresponding to the partition. The local modes are extracted from each sub-snapshot. Discussion and notation are shown in Sect. 3.3; (iii) the entire model is assembled by a few (or a single) generic parts. The global solutions of each archetype are stacked into separate datasets, and the RBs are constructed with regard to each reference shape.

The so-called *localized training* strategy is utilized for generating subdomain-level FOM solutions, in which independent simulations are carried out in a small system. The computational domain consists of several generic blocks, which are much smaller than the original model. Thus, the cost of the offline stage is significantly reduced. We highlight that the technique assumes local features still accurately represent the total dynamics. *Subdomain-specific solutions* are collected as snapshots. A RB is computed for each archetype block and then is transformed into the instantiations, as explained in Sect. 3.3.

2.4 Acceleration Techniques

The offline stage of ROMs can be computationally expensive, especially for large-scale problems with wide parameter ranges. The online stage can likewise be costly for nonlinear problems. Consequently, various acceleration techniques have been developed for both stages. Offline acceleration often uses hybrid strategies that combine the accuracy of FOMs with the efficiency of ROMs (see Sect. 3.4.1). Online acceleration uses hyper-reduction methods to lower the cost of evaluating nonlinear terms in the reduced system, as shown in Sect. 3.4.2. These techniques are applicable to projection-based and data-driven frameworks, and to both global and local approaches.

2.5 Coupling Techniques

Once the local RBs are available, the next step is to *glue* them to approximate global solutions. This coupling has gained attention from the research community in recent years. Therefore, various methods have been developed to achieve it. We highlight that the fundamental philosophy of the algorithms is to minimize discontinuities at the adjacent interfaces while satisfying the governing equations and boundary (and initial) conditions.

Similar to standard ROMs, local ROM approaches can be constructed employing *projection-based* (Sect. 4) or *data-driven* (Sect. 5) methodologies.

Regarding how reduced systems are constructed and solved, projection-based approaches can be classified into two groups: (i) *monolithic* schemes (Sect. 4.2) and

(ii) *iterative* algorithms (Sect. 4.3). The two groups are described in Sect. 4.

Data-driven techniques are becoming popular due to the rapid development of AI. Moreover, it is a non-intrusive methodology that reduces the complexity of handling PDEs, enhancing the framework's adaptability to different governing equations. Recently, various interpolation/regression algorithms, as well as *Neural Networks*, have been intensively employed for constructing local ROMs (see Sect. 5).

Given the diversity of projection-based and data-driven coupling methods, two flowcharts regarding their classifications are presented in Sections 4 and 5, respectively. We also note that separate descriptions for multiphysics problems are involved, due to their specific characteristics in coupling.

In short, the procedure and classification for constructing local ROMs are illustrated in Fig. 1. The three preliminaries are carried out first, and then either projection-based or data-driven coupling techniques are applied to reconstruct global fields.

3 Preliminaries

Now, we will explain the four preliminaries: (i) decomposition, (ii) parameterization, (iii) snapshot collection and dimensionality reduction at the local level, and (iv) acceleration techniques for the offline and online stages. Before discussing these four preliminaries, we first clarify the notation used in this review. The rules are listed in Table 1.

3.1 Domain Decomposition

Domain decomposition has been widely used in high-fidelity simulations for parallel computing and multi-physics problems. In a well-known book, Quarteroni et al. [52] discussed

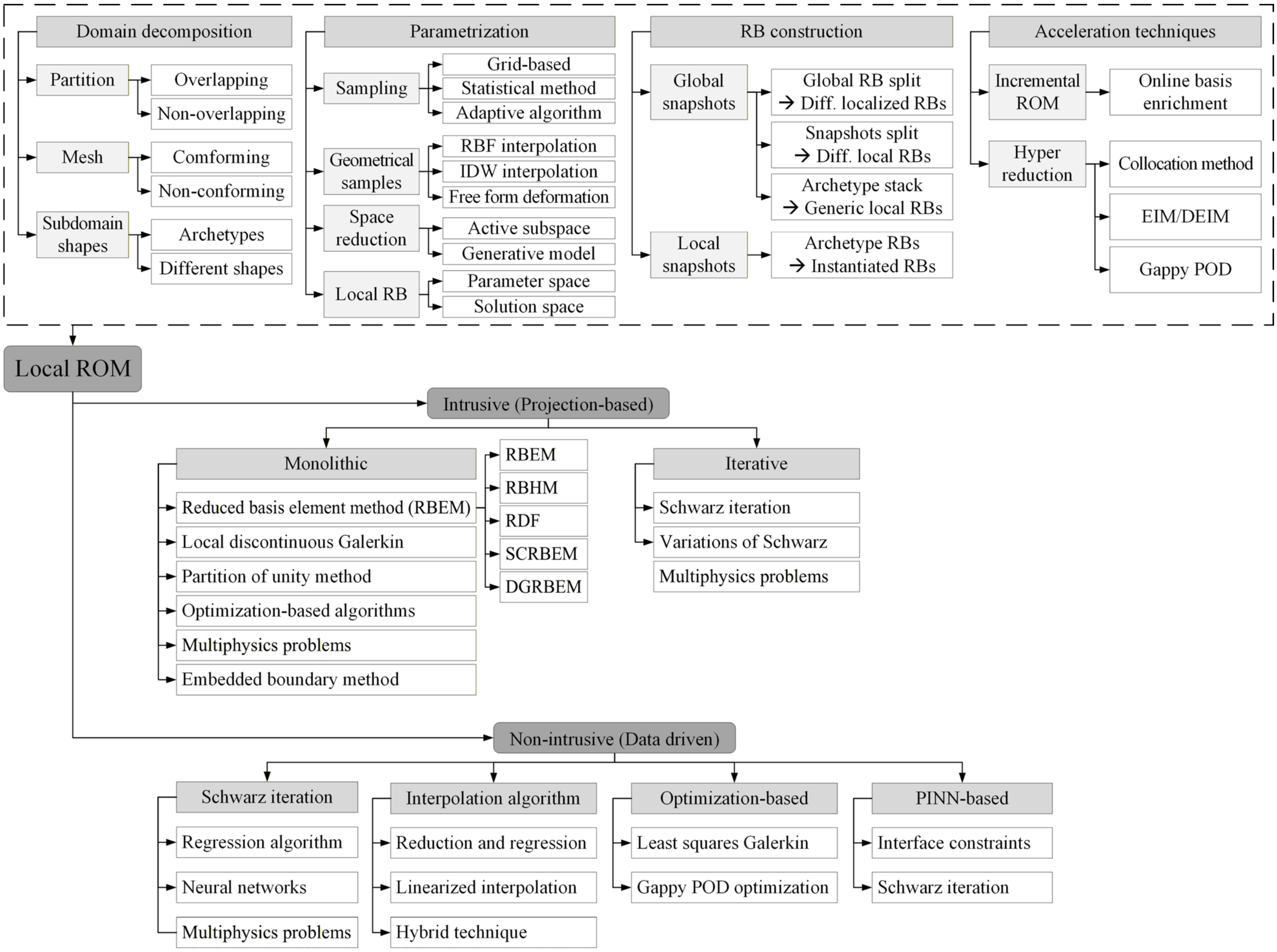
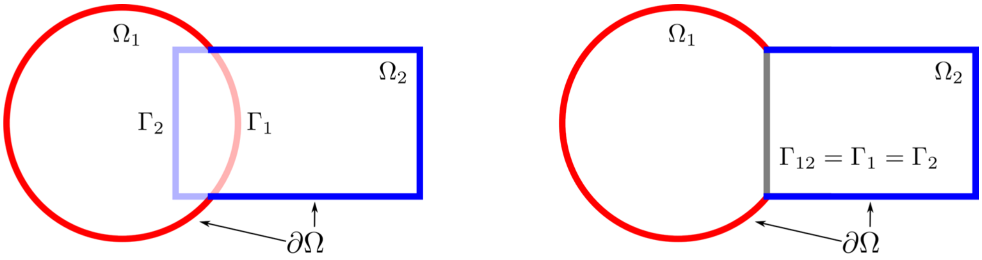


Fig. 1 Four preliminaries for constructing ROMs and the categorization of intrusive and non-intrusive coupling techniques. Abbreviation: RBF (radial basis function), IDW (inverse distance weighting), EIM (empirical interpolation method), DEIM (discrete empirical interpolation method), RBHM (reduced basis hybrid method), RDF (reduced basis, domain decomposition, and finite element method), SCRBEM (static condensation reduced basis element method), and DGRBEM (discontinuous Galerkin reduced basis element method)

Table 1 Notation rules used throughout this paper

Rule	Description	Examples
Scalars and vectors	Scalars, scalar and general fields and are denoted by plain italic symbols, whereas vectors and vector fields are denoted by bold symbols.	Scalar field $u(\mathbf{x};\boldsymbol{\mu})$, vector field $\mathbf{u}(\mathbf{x};\boldsymbol{\mu})$, spatial coordinate $\mathbf{x}$, and parameter vector $\boldsymbol{\mu}$.
Operators	Continuous, mapping, and surrogate operators are underlined.	PDE operator $\underline{Q}$, geometric transformation $\underline{T}$, and surrogate operator $\underline{Z}$.
Algebraic terms	Vectors and matrices in the algebraic system are double-underlined. Their entries are not underlined; for example, M_{ij} denotes the (i,j)-th entry of $\underline{\underline{M}}$.	Snapshot matrix $\underline{\underline{S}}$, and algebraic matrix $\underline{\underline{M}}$.



(a) Overlapping decomposition (b) Non-overlapping decomposition

Fig. 2 Schematic of domain decomposition: (**a**) overlapping; (**b**) non-overlapping. The global domain Ω is divided into Ω_1 and Ω_2 parts with boundary $\partial\Omega$. For overlapping, two interfaces exist, Γ_1 and Γ_2; for non-overlapping, a common interface $\Gamma_{[12]}$ exists

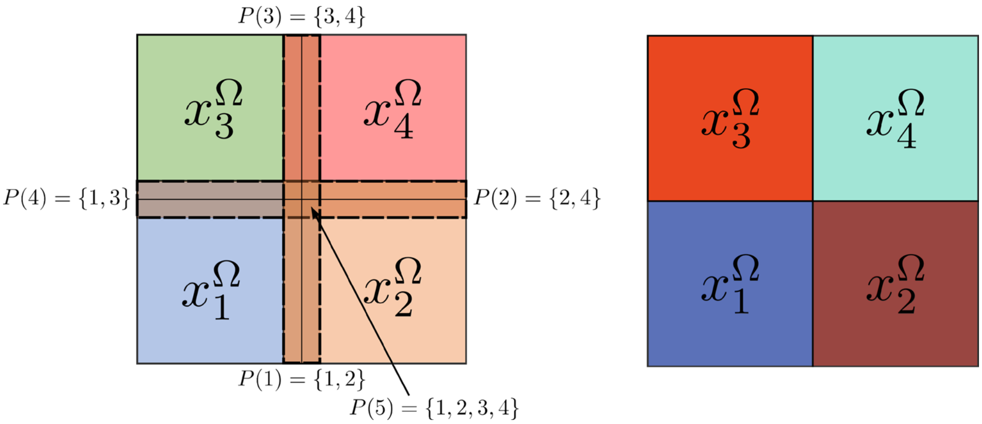


Fig. 3 Domain decomposition into 2×2 subdomains. Left: overlapping partition, redrawn based on [54]. Right: non-overlapping partition, redrawn based on [37]

the fundamental topics of the theory, numerical analysis, and implementation of domain decomposition approaches.

In the following paragraphs, we focus on the implementation of domain decomposition procedures corresponding to the local ROM, and the three classification aspects are presented with descriptions and sketches.

3.1.1 Decomposition With/Without Overlaps

There are two main types of partitions: with or without overlaps. The decomposition strategy primarily depends on the chosen coupling algorithms, which are discussed in later sections. Here, we briefly illustrate these two partition types.

The overlapping and non-overlapping decompositions are schematized in Fig. 2. We assume a domain Ω and its boundary $\partial\Omega$. Ω is divided into pieces Ω_1 and Ω_2. For the overlapping condition, this division results in two interfaces Γ_1 and Γ_2. The non-overlapping procedure creates a shared face $\Gamma_{[12]}=\partial\Omega_1\cap\partial\Omega_2$[3]. For clarity, we note Ω as a n-dimensional domain, where $n=2,3$. The closure of the domain is denoted by $\overline{\Omega}=\Omega\cup\partial\Omega$.

When the geometry is split into multiple partitions, two adjacent subdomains Ω_m and Ω_n with overlaps have interfaces denoted by $\Gamma_{[mn]}=\partial\Omega_m\cap\Omega_n$ and $\Gamma_{[nm]}=\partial\Omega_n\cap\Omega_m$. Nevertheless, for non-overlapping scenarios, a single common internal face is noted $\Gamma_{[mn]}=\partial\Omega_m\cap\partial\Omega_n$.

We claim that both methodologies have distinct advantages and limitations. Increasing the shared regions enhances the stability and convergence of coupling techniques [53]. However, overlaps introduce challenges in partitioning. Fig. 3 (left) illustrates a 2×2 domain decomposition, where the overlapping relationships become increasingly complex [54]. In particular, the central subregion represents the interaction of all four neighboring subdomains, significantly complicating the construction of a ROM.

---

[3] The subscript $_{[12]}$ is a combined index to designate a single face.

In contrast, the non-overlapping approach offers greater flexibility and adaptability, simplifying the subdomain generation process. As depicted in Fig. 3 (right), interfaces exist only between adjacent subdomains, reducing geometric complexity. This approach also allows more flexibility for mesh discretization, where adjacent subdomains can employ either conforming or non-conforming meshes [23].

Furthermore, a shared region implies that the same physical phenomena probably occur in the adjoining subdomains [25]. However, non-overlapping techniques enable the coupling of subdomains with different governing PDEs. Despite this advantage, the simplicity of the non-overlapping approach does not come for free. The approaches are currently limited by the complexity of developing robust algorithms for coupling subdomains at the connected interfaces [55].

3.1.2 Conforming and Non-Conforming Meshes

The conforming and non-conforming high-resolution meshes along an interface are displayed in Fig. 4. This characteristic significantly affects the algorithms for coupling subdomains. In some cases, additional mutual interpolation procedures are required for the reconstruction of global solutions [39].

3.1.3 Individual and Generic Decomposition

As indicated in Sect. 1, a global model can be decomposed into many independent subdomains or assembled by a few types of generic components (archetypes). The use of these approaches depends on the geometric characteristics and the coupling algorithms. We will clarify the two strategies here.

3.1.3.1 Individual Decomposition *Individual decomposition* denotes a domain Ω composed of a finite number of open and bounded partitions,

$$\Omega = \bigcup_{m=1}^{N_\Omega} \Omega_m,$$

where Ω_m is the m^{th} subdomain, and N_Ω is the total number of partitions. The interfaces between adjacent subdomains are denoted by Γ_r, where $r = 1, \ldots, N_\Gamma$. The global boundary of Ω is assigned as $\partial\Omega$, and it holds that $\partial\Omega \cap \Gamma = \emptyset$. The local boundary of Ω_m is similarly noted as $\partial\Omega_m$, and we have $\partial\Omega \cap \Gamma \neq \emptyset$.

Indeed, the interfaces coincide with the specific boundaries of the partitions. Generally, a partition Ω_m connects to several interfaces Γ_r. Thus, we define the set of interfaces connected to Ω_m as

$$\mathcal{I}_m = \{\Gamma_r \,|\, \Gamma_r \subset \partial\Omega_m\}.$$

The number of interfaces connected to Ω_m is given by $N_m = |\mathcal{I}_m|$. The i^{th} item in $\mathcal{I}_m$ is noted as $\Gamma_{m,i}$. Note that we also use the notation $\Gamma_{[mn]}$ to denote the interface between two adjacent subdomains Ω_m and Ω_n. In this case, $\Gamma_{[mn]}$ is an element of both $\mathcal{I}_m$ and $\mathcal{I}_n$. The two notations $\Gamma_{[mn]}$ and $\Gamma_{m,i}$ are used to identify the interfaces, but the former is more convenient for describing the interaction between two adjacent subdomains, while the latter is more suitable for describing the relationship between a partition and its connected interfaces.

For domains without symmetry and periodicity, the splitting results in subdomains Ω_m of various shapes. Fig. 5 displays two examples, i.e., laminar flow around a cylinder [56] and pollutant transport within a city [57]. Furthermore, if the model contains subregions governed by different PDEs, the entire geometry is divided considering the physical boundaries. The FSI problems are typical instances consisting of both fluid and solid partitions. Two FSI studies are presented in Fig. 6, which illustrate the interactions of a linear elastic structure and a fluid flow governed either by Stokes or incompressible Navier-Stokes equations.

3.1.3.2 Generic Decomposition The second strategy is called *generic decomposition*. It is named to contrast with the previous *individual* strategy. Assume we have $\hat{N}_\Omega$ reference subdomains, and the k^{th} of them is denoted by $\hat{\Omega}^k$. We use the *hat* symbol $\hat{\square}$ to characterize reference domains. Each archetype $\hat{\Omega}^k$ has N_k instantiations, indexed by

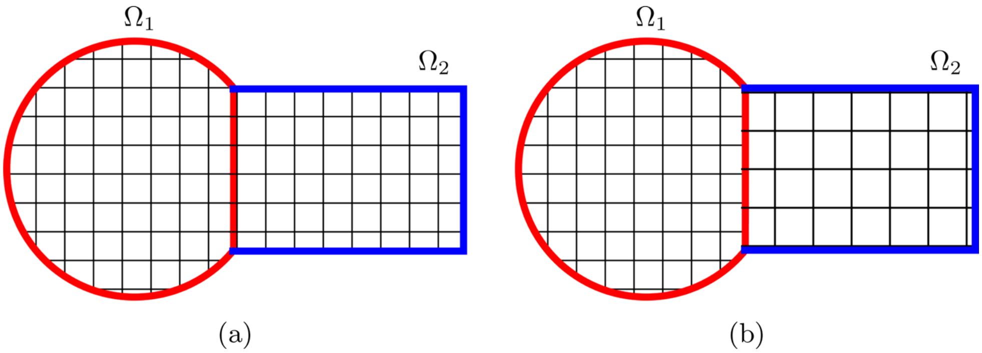
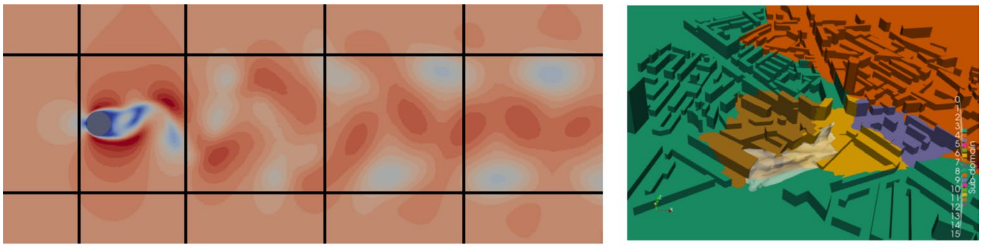


Fig. 4 (**a**) Conforming and (**b**) non-conforming meshes for adjacent subdomains



(a) (b)

Fig. 5 Examples of domain divisions for problems without repeating geometric parts. (**a**) Kármán vortex street for flow around a cylinder. A different number of POD modes can be used in each subdomain to capture the flow features, e.g., 3 modes for the upstream region, 12 modes for the vortex region, and 6 for the rest. Figure is redrawn based on [56]. (**b**) Pollutant transport in an urban environment, taken with permission from [57], copyright owned by IOS Press

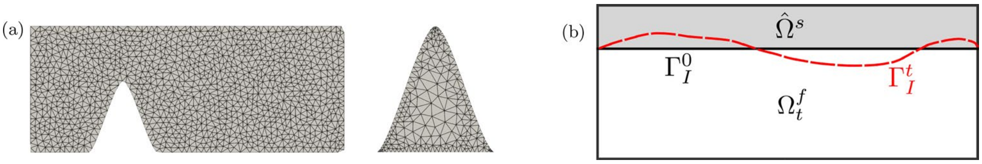


Fig. 6 Decomposition for fluid-structure interaction problems. (**a**) Left:: fluid; right: solid. Redrawn based on [37]. (**b**) Solid is denoted by $\hat{\Omega}_s$ and Ω_t^f is fluid. Γ_I^0 and Γ_I^t denote the initial and time-dependent interfaces positions, respectively. Redrawn based on [58]

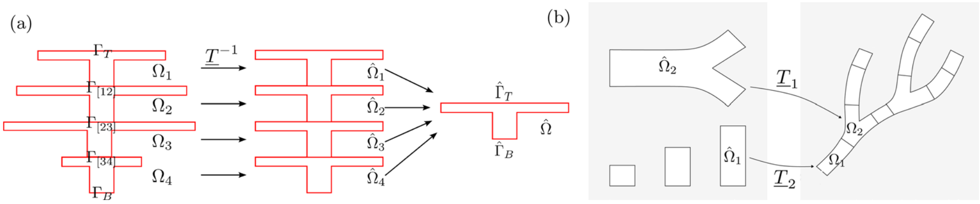


Fig. 7 Several generic parts can be used to assemble entire models. The real blocks are obtained by geometric transformation from reference parts. (**a**) a multi-level thermal fin. Both interfaces and subdomains are mapped back to the archetypes, e.g., $\Omega_1 \circ \underline{T}_1^{-1} = \hat{\Omega}$ and $\Gamma_{[12]} \circ \underline{T}_{[12]}^{-1} = \hat{\Gamma}_B$. Redrawn based on [59]. (**b**) a blood vessel formed from blocks, e.g., $\Omega_1 = \hat{\Omega} \circ \underline{T}_1$ and $\Omega_2 = \hat{\Omega} \circ \underline{T}_2$, taken with permission from [33], copyright owned by Elsevier

$m = 1, \ldots, N_k$, namely Ω_m^k. These instances are obtained through a parameterized geometrical transformation $\underline{T}^k(\boldsymbol{\xi}_m^k)$[4], namely,

$$\Omega_m^k := \underline{T}^k(\boldsymbol{\xi}_m^k) \circ \hat{\Omega}^k, \tag{1}$$

where $\boldsymbol{\xi}_m^k$ is the geometric transformation parameter for Ω_m^k. We use vector notation to represent $\boldsymbol{\xi}_m^k$, as they can be multi-dimensional.

We also note the total number of partitions with N_Ω, and obviously, it holds that $N_\Omega = \sum_{k=1}^{\hat{N}_\Omega} N_k$. Consequently, the entire computational domain is expressed as

$$\Omega = \bigcup_{k=1}^{\hat{N}_\Omega} \bigcup_{m=1}^{N_k} \Omega_m^k,$$

and we define the set of interfaces connected to Ω_m^k as $\mathcal{I}_m^k$.

Many realistic industrial facilities are designed to be periodic and symmetric, as these characteristics enhance process consistency, efficiency, and cost-effectiveness. Such structural regularity can be exploited to simplify simulations and significantly reduce computational costs at both FOM and ROM levels. The thermal fin pattern displayed in Fig. 7 is a typical design for heat exchangers [60]. In [59], the authors analyzed the heat transfer in a multiple-level thermal fin, in which each block is transformed from a generic part. A similar concept appears in biological modeling, as presented in [33], where the blood vessel structure is composed of four reference blocks. The applications demonstrate that a few archetypes can assemble even complex flow networks.

---

[4] We use the underline symbol $\underline{\square}$ to characterize operators, and $\underline{T}^k$ denotes the transformation operator.

3.2 Parameterization Techniques

ROMs are constructed from high-resolution datasets within a specific parameter range. Hence, the choice of parameters is crucial for the performance of the ROM. Below, we discuss four aspects of parameterization, which are sampling strategies, geometrical parameterization, parameter space reduction, and local reduced bases in the parameter (or solution) space. The first two concern the selection of physical and geometrical parameters, while the last two address very high-dimensional parameter spaces.

3.2.1 Sampling Strategies

Sampling consists of selecting a subset of the observed data, which can be used to estimate the features of the entire dataset [61]. It is important for applications like design optimization and uncertainty quantification. The theory of general sampling is extensive and beyond the scope of this work (see [61, 62] for details). Thus, we only outline studies relevant to ROMs for the sake of completeness. Interested readers can also turn to [12, 63, 64] for a relatively concise summary of the topic.

The choice of sampling strategies depends mostly on the dimension, $N_{\mathcal{P}}$, of the parameter space $\mathcal{P}$ [63]. *Grid-based* approaches are the standard option for problems with low dimensions (typically $N_{\mathcal{P}} \leq 3$) [12, 65]. For example, three grid-based techniques for $\mathcal{P} = [-1, 1]^2$ are plotted in the first row of Fig. 8. Non-uniform selections, such as *Clenshaw-Curtis* points [66], are an alternative. By employing these strategies, one can choose the same or different numbers of values in each dimension.

When more parameters are considered, the number of grid points grows exponentially. Statistical algorithms like random (e.g., *Monte Carlo*) and *latin hypercube* sampling [67], as well as Smolyak sparse grid, are well-suited for these conditions [12, 65] (see second row of Fig. 8). One can also employ a probabilistic sample method based on centroidal Voronoi tessellations [68].

The situation becomes more challenging when the dimension $N_{\mathcal{P}} > 10$. The common sampling techniques mentioned above are not feasible in these conditions. Namely, it is impossible to balance numerical cost with a good representation of $\mathcal{P}$. Therefore, more sophisticated problem-specific *adaptive algorithms* should be utilized, as indicated in [63–65].

The greedy algorithm is a widely used adaptive sampling strategy for ROMs. It iteratively selects the parameter in the candidate set that maximizes the error of the current ROM, and then enriches the reduced basis with the solution corresponding to this parameter. This process continues until a desired accuracy is achieved. This strategy can optimize the number of high-fidelity simulations, which is the bottleneck of the offline stage.

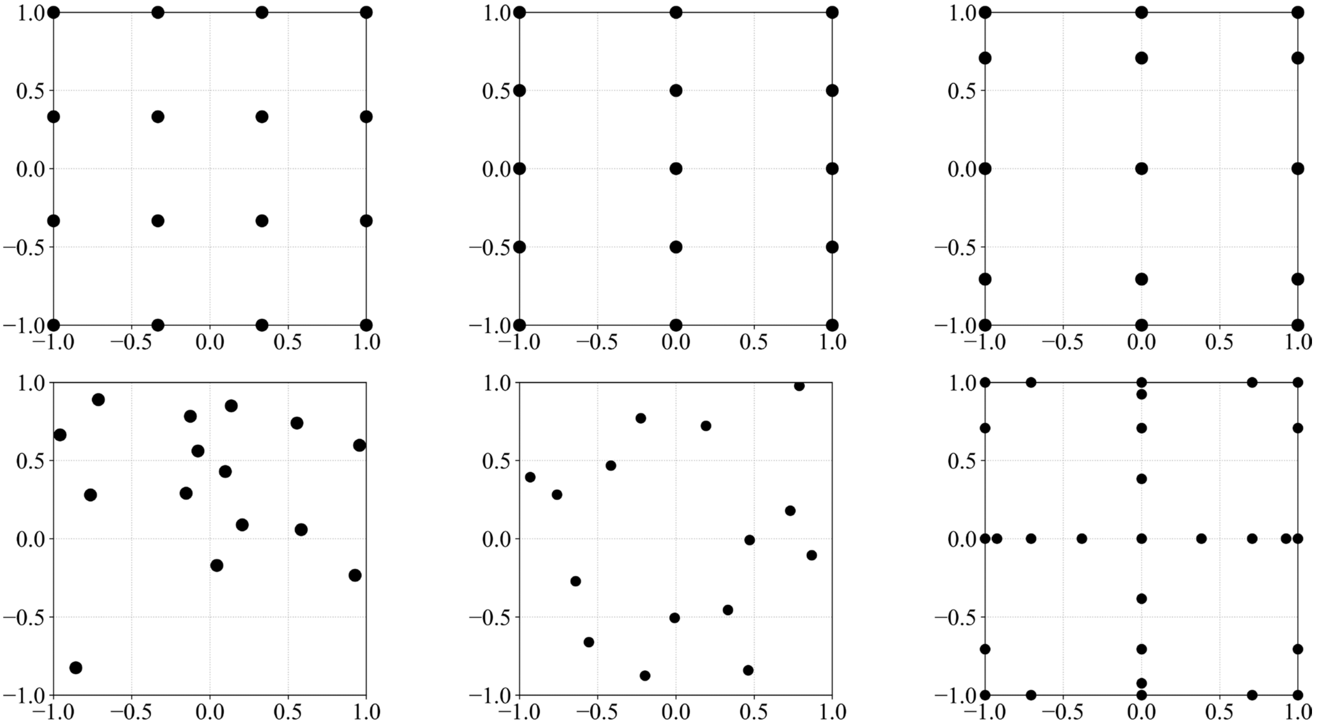


Fig. 8 Sampling strategies in the space $\mathcal{P} = [-1, 1]^2$. Grid-based (first row, from left to right): 4 × 4 uniform, 3 × 5 uniform, and 3 × 5 Clenshaw-Curtis. Statistical (second row, from left to right): random (15 points), latin hypercube (15 points), and smolyak sparse grid with Clenshaw-Curtis points. Figures redrawn based on [12]

Greedy algorithms are generally used in *intrusive* ROMs. Several early-stage developments include the studies of Patera and Maday et al. [69–72] and Haasdonk et al. [73]. Amsallem et al. [74] developed an adaptive POD-Greedy approach, which uses a surrogate model to identify a set of poorly represented parameters first, and then performs reduced simulations to select the new parameter to be enriched. Some recent studies can be found in [75, 76]. Additionally, the usage of the greedy algorithm in non-intrusive ROMs can be found in [77, 78].

Note that the error in the iterative process of the greedy algorithm is estimated by an *error estimator*, which is derived for different types of PDEs, such as elliptic [79, 80], parabolic [81], Stokes [82], and Navier-Stokes equations [83].

There are also several alternative adaptive approaches, such as local sensitivity analysis [84], the trust region algorithm [85], a gradient-based method [86], and isomap-based approaches [87–89]. The performance of different adaptive algorithms for the application of the ROM is compared in [64, 90].

Recently, *active learning* has gained popularity for parameter sampling and applications in ROMs. Active learning is a general machine learning framework that actively selects samples to maintain the performance of the current model [91]. The selection can be achieved through various optimization strategies. The aforementioned greedy algorithm can be integrated into the active learning framework [91]. In [92], the authors used a neural network to predict the error of the current ROM, and then, identify samples with the highest predicted errors. Zhang et al. used the probability density function of ROM errors as a criterion to adaptively advance the sampling process [93].

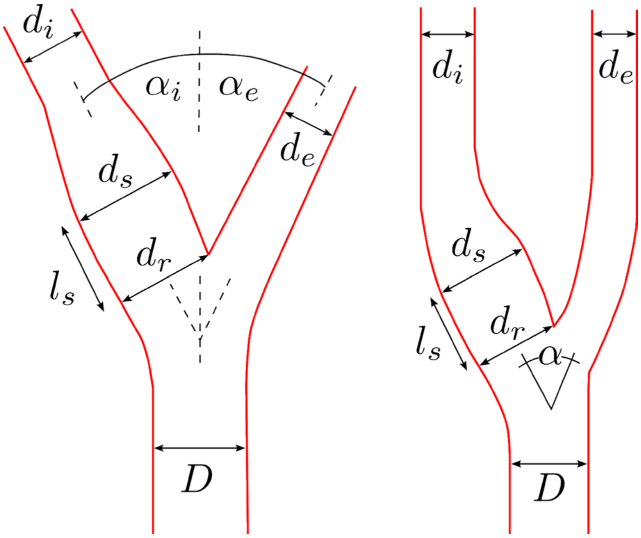


Fig. 9 Parameters for a carotid artery bifurcation [107–109]. The shape depends on channel diameters D, d_i, d_s, d_r, d_e and bifurcation angles α_i, α_e. Figures redrawn based on [110]

Moreover, we emphasize that *Bayesian optimization* is widely incorporated with active learning for sampling [94]. Bayesian optimization is a probabilistic technique that aims to find the global optimum of functions that are expensive to compute. This makes it well-suited to the active learning framework for estimating the performance of the current ROM and selecting new samples to improve accuracy. Applications of the algorithm for building ROMs can be found in [94–98]

3.2.2 Geometrical Parameterization

Geometrical parameterization is important in industrial practice for design and optimization. See descriptions and examples in [12, 99, 100]. In terms of implementation, it is worth mentioning the Python package PyGem [101], which employs various techniques for generating geometrical samples. For the numerical analysis of propeller blades in particular, the geometrical parameterization and bottom-up construction can be performed using BladeX (Python Blade Morphing) [102].

Mostly, geometrical parameterization aims to achieve two goals: (i) shape optimization; (ii) efficient simulation of multiple shapes. The former is studied for many applications, including aircraft wings [103], marine propellers [104], and ship hulls [105].

In the other condition, solutions for numerous deformed geometries are required. For instance, in cardiovascular and blood system simulations, the shape of the computational domain is patient-specific [? [33].

The procedure for geometrical parameterization is not unique. Different methods have been developed to meet specific requirements. For discussion, we identified two subcategories: *interpolation technique* and *free form deformation*.

3.2.2.1 Interpolation Technique The general procedure for generating samples via interpolation can be summarized in three consecutive steps: (i) selecting parameters; (ii) deforming a reference geometry with regard to different parameters to generate a series of shapes; (iii) obtaining meshes from the reference one by interpolating the deformation of shapes. To clarify, we will provide three examples.

Firstly, we regard the simple thermal fin problem displayed in Fig. 7a, in which the fins differ only in their length [59, 106].

Secondly, we consider the complex model of Fig. 9 that shows the parametric geometry of the human carotid artery bifurcation [107–109]. To characterize the domains, the

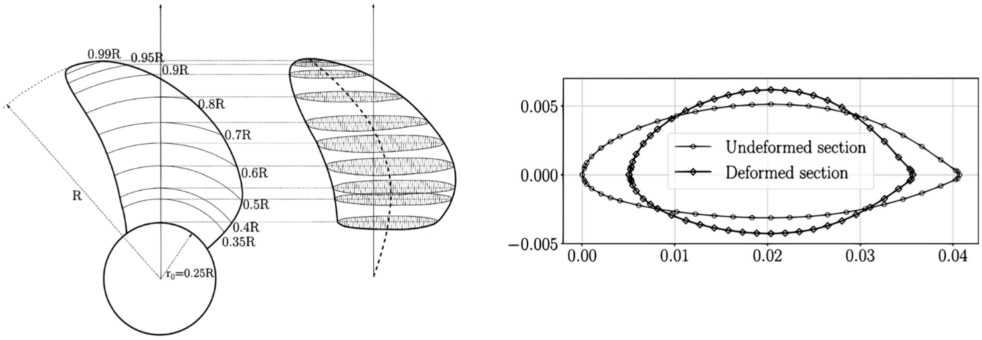


Fig. 10 Definition of the parameters of a marine propeller's blade. The blade is characterized by a series of sections at different radius values (right), and each piece can be deformed (left). Figures taken with permission from [104], copyright owned by John Wiley & Sons, Inc

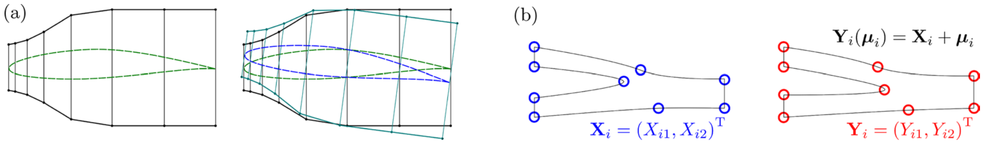


Fig. 11 Generation of geometric samples by the displacements of control points. (**a**) An aircraft wing, redrawn based on [113] and (**b**) a bypass problem with the initial point location $\mathbf{X}_i$, and the deformed point coordinate $\mathbf{Y}(\boldsymbol{\mu}_i) = \mathbf{X}_i + \boldsymbol{\mu}_i$, redrawn based on [110]

authors defined several specific quantities, i.e., flow channel diameters, branch curvatures, and bifurcation angles.

Lastly, a similar strategy is proposed by Ivagnes et al. [104] for the optimization of marine propellers. The parameters are assigned at the level of a blade section, and thus, they determine the entire propeller.

Once we obtain shape samples, we can compute the meshes via interpolation. *Radial Basis Function* (RBF) interpolation is a widely used technique. Let us briefly describe it. Assume that input variables x_i and results y_i, with $i = 1, \dots, N$ are known. Our objective is to approximate the function $\underline{f}$, achieving $y_i = \underline{f}(x_i)$. In this technique, the interpolation can be written as

$$y_i = \sum_{j=1}^{N} w_j \underline{\phi}(\|x_i - x_j\|) \quad \text{with} \qquad j = 1, \dots, N,$$

where $\underline{\phi}$ denotes a RBF that is weighted by coefficients w_j, $j = 1, \dots, N$, and the unknowns w_j are computed by solving the equation with N pairs of x_i and y_i.

Once the RBF interpolation model is trained, it can be applied to estimate y corresponding to the numerous new x. The definition of $\underline{\phi}$ is not unique. More details are shown in [111, 112].

Note that the cases illustrated above employ problem-specific configurations, which are confined to individual applications. Now, we introduce a general strategy. This generates the geometrical shapes by the position of certain control points [110, 113]. The ideology is plotted in Fig. 11. The set of control points is defined either surrounding or on the surfaces/edges of the model. The translations of points are regarded as parameters. The deformed outer face and mesh can be interpolated using RBF.

Note that the geometric interpolation techniques can also be incorporated to approximate the mesh motion in FSI problems. For example, the usage of RBF is presented in [114]. *Inverse Distance Weighting* (IDW) is an alternative to RBF. The results in [115, 116] demonstrated that IDW is comparable in accuracy to RBF and accelerate the high-fidelity simulations. The combination of a ROM and IDW is presented in [114, 117]. However, the interpolation approaches become costly for numerous points. Thus, an adaptive selective RBF method is presented in [114], and the *Selective IDW* (SIDW) algorithm is proposed in [117, 118]. The fundamental idea of the two variants is the same, in which only a subset of points is optimally identified for performing interpolation. See more details about these investigations in Section 4.2.7.

3.2.2.2 Free Form Deformation We will introduce another general and simple method that is well-suited for the geometrical transformation of two-dimensional (2D) and three-dimensional (3D) objects, namely the *FFD*. Note that the FFD also involves the movement of control points. We will now outline the procedure and indicate the differences compared to previous interpolation techniques.

The procedure of FFD is sketched in Fig. 12. The FFD transformation $\underline{T}(\cdot, \boldsymbol{\mu})$ consists of three operations

$$\underline{T}(\cdot, \boldsymbol{\mu}) = \underline{\Psi}^{-1} \circ \hat{\underline{T}}(\cdot, \boldsymbol{\mu}) \circ \underline{\Psi},$$

where $\underline{\Psi}$ is the mapping from the physical domain to a unit cube, $\underline{\Psi}^{-1}$ is the inverse mapping, and $\hat{\underline{T}}(\cdot, \boldsymbol{\mu})$ is the deformation of the unit cube. The three steps are detailed as follows.

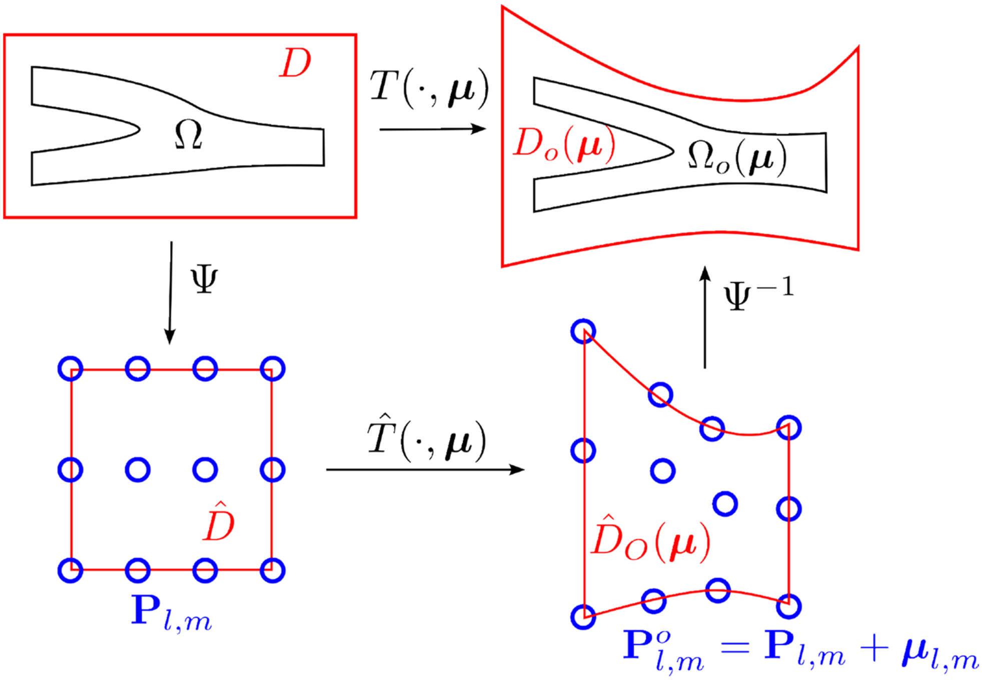


Fig. 12 Two-dimensional FFD implementation for a bypass structure. FFD deforms a lattice (red) via moving the grid control points (blue). The new shape is obtained using mapping and interpolation. Figure redrawn based on [119]

Firstly, as shown in Fig. 12, a reference bypass structure, Ω, is embedded in a rectangle, D. By applying transformation $\underline{\Psi}$, D is mapped into a unit cube $\hat{D} \equiv [0,1]^2$. We consider L and M control points inside $\hat{D}$. L and M are defined on the abscissa and ordinate axes, respectively. Each grid point is defined by its two coordinates and indexed by

$$\mathrm{P}_{l,m} = [l/L, m/M],$$

where l and m are the indices of the control point, with $l = 0, \ldots, L$ and $m = 1, \ldots, M$.

Secondly, the transformation $\underline{\hat{T}}(\cdot, \boldsymbol{\mu})$ is defined. Assume the displacement of each point is $\boldsymbol{\mu}_{l,m}$, so the deformed control point is given by $\mathrm{P}^o_{l,m} = \mathrm{P}_{l,m} + \boldsymbol{\mu}_{l,m}$. $\hat{D}$ is deformed to $\hat{D}_o$ by interpolating the new control point positions. For any coordinate $\hat{\mathbf{x}} \in \hat{D}$, the new position is calculated with the following transformation:

$$\underline{\hat{T}}(\hat{\mathbf{x}}, \boldsymbol{\mu}) = \sum_{l=0}^{L} \sum_{m=1}^{M} b^{L,M}_{l,m}(\hat{\mathbf{x}}) \mathrm{P}^o_{l,m},$$

where $b^{L,M}_{l,m}(\hat{\mathbf{x}})$ denotes tensor product of the 1D Bernstein basis polynomials (see more in [120]).

Thirdly, the inverse mapping is applied, namely, $D_o = \underline{\Psi}^{-1} \circ \hat{D}_o$, and thus, the deformed shape Ω_o is obtained.

FFD has become popular due to several valuable characteristics [120]: (i) It is suitable for surfaces of any formulation or degree; (ii) Its implementation is independent of the domain and mesh; (iii) It can be applied locally or globally and preserves smoothness when incorporating deform only for subregions.

There are many relevant studies about the combination of a ROM and FFD discussing different phenomena: shape parameterized heat conduction problem [121]; the immersed bodies in Stokes flows [122]; PDE-constrained optimization problems in haemodynamics [123]; unsteady Navier-Stokes flow inside the Coronary artery bypass [124]; linear/nonlinear constrained design optimization of ship hull (governed by Navier-Stokes equations) [125]. Another highlighted investigation is addressed by Forti [114], in which he adopted RBF, IDW, and FFD for FSI scenarios.

The FFD can also be applied locally. Fig. 13 shows a simulation of Navier-Stokes flow in a shrinking cardiovascular structure [126]. Another study for the aerodynamic shape optimization of a car's front end is presented in [103]. Forti et al. successfully integrated the domain decomposition approach with FFD for the simulation of a fluid-structure interaction problem [114]. The deformation is performed in each subdomain, and thus, local control point refinement can be achieved in specific regions.

3.2.3 Parameter Space Reduction

The aforementioned techniques cover the general strategy for generating samples, including physical and geometrical. However, parameterization may become problematic when associated with a high-dimensional space, which results in difficulties in design optimization, sensitivity analysis, uncertainty quantification, etc.

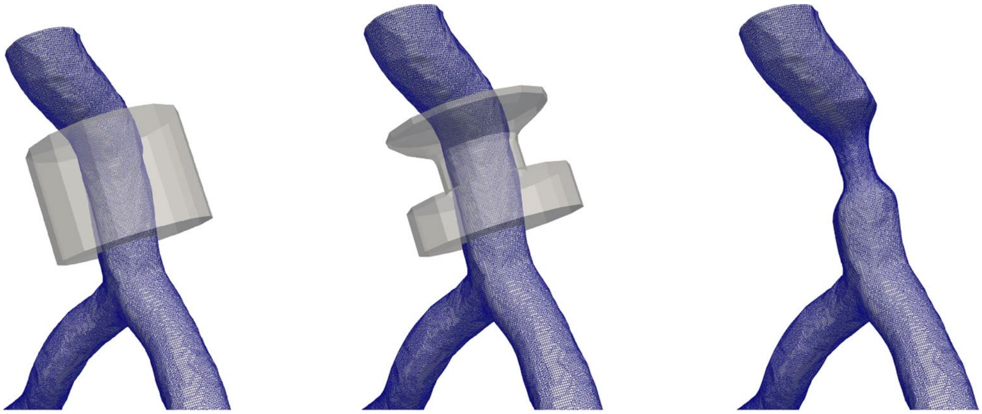


Fig. 13 FFD for local deformation of a cardiovascular structure. e.g., a blood vessel. Figure taken with permission from [126]

Some examples may be provided to clarify this issue. Suppose that shape optimization is performed employing FFD with a large amount of control points. The displacements of every point should be sampled, which is not practical for the design workflow of realistic applications. To deal with these difficulties, the *parameter space reduction* technique arose. In summary, it involves moving points in a correlated manner, specifically by adjusting a few *synthetic* parameters, and then obtaining samples based on this reduction.

Several studies have shown the key benefits of parametric reduction. Clearly, fewer inputs reduced the cost of the online stage, which can accelerate the analyses of sensitivity and uncertainty [127] with ROMs. Additionally, it was shown that low-dimensional inputs also improve the accuracy of ROMs [125, 128–130].

For completeness of the section, we will illustrate two popular techniques regarding parametric reduction in ROMs: *Active subspaces* and the *AI-based generative model*.

3.2.3.1 Active Subspaces The *Active Subspaces* (AS) method has recently been used for linear reduction in input spaces [130]. It is a dimensionality reduction technique for high-dimensional spaces that identifies the most influential subspace of the inputs with respect to an output of interest. By employing AS, one can use linear combinations of the original parameters as new composite variables, instead of the original individual parameters. This allows for a more efficient process of sampling and optimization. The steps and mathematical details of the method are presented in Appendix B.1.

The incorporation of ROMs and AS has been tested in various scenarios. Demo et al. [128] revealed that their non-intrusive ROM involving AS outperforms the one disregarding this technique. The framework has also been utilized intensively for shape design and optimization. The applications include airfoil [129], ship hull [125], and ship propeller blade [131] optimizations.

We remark on a recent publication [130], a slight variation of the method. There, the authors managed to partition the parameter space with clustering algorithms. Their method, the so-called *local active subspace* procedure, results in a more efficient and accurate dimension reduction than the standard one.

3.2.3.2 Generative Model AI-based methods have been employed to generate 3D objects in computer vision. This success suggests the feasibility of utilizing them to create geometrical samples. Given their relevance, we start introducing data-driven *generative models* for parameter space reduction [132].

As mentioned before, FFD is a flexible and general technique for the generation of geometries. However, when considering linear/multi-linear constraints, its numerical cost can not be ignored.

Thus, Padula et al. [132] proposed a strategy based on neural networks to generate efficiently constrained samples. It is composed of two parts: an Encoder $\underline{E} : \mathbb{R}^N \to \mathbb{R}^n$ and a decoder $\underline{D} : \mathbb{R}^n \to \mathbb{R}^N$, in which $n \ll N$. It achieves the approximation that $\boldsymbol{\mu} \approx \underline{D} \circ \underline{E}(\boldsymbol{\mu})$. The vector $\underline{E}(\boldsymbol{\mu}) \in \mathbb{R}^n$ is the latent representation of $\boldsymbol{\mu}$, and $\mathbb{R}^n$ is the latent space.

The implementation of the generative model consists of three steps: (i) a few samples generated by constrained FFD are applied to train the network; (ii) instead of $\boldsymbol{\mu}$, the vector $\underline{E}(\boldsymbol{\mu})$ is adopted; (iii) new samples are produced utilizing decoder $\underline{D} \circ \underline{E}(\boldsymbol{\mu})$. The procedure can utilize a simple *Autoencoder* (see Sect. 5.1.2) or more complex frameworks, which were described and compared in [132].

This model has been validated utilizing several benchmark cases, demonstrating its effectiveness across different applications. These include the Poisson problem in Stanford Bunny geometry and Reynolds Averaged Navier-Stokes equations for the shape optimization of a naval hull. These tests have revealed that the method can represent constraints efficiently and result in a more accurate ROM [132]. Notably, both the *active subspace method* and the *generative model* confirm the significant benefits of exploiting parameter space reduction techniques.

3.2.4 Local Reduced Basis for Clustered Datasets

Here, we will introduce the concept of the *local RB* for clustered datasets. The idea is to partition the parameter space or solution space into several subgroups and compute a local RB for each subgroup. This approach can improve the accuracy of the ROM when the parameter space is large and complex. The approach is proposed to tackle the issue of the performance deterioration of the ROM when using a single RB for a wide range of parameters. Note that these RBs are localized in the parameter/solution space, which is different from the local RBs defined spatially in the subdomain-level (discussed in Sect. 3.3).

The clustering treatment can be regarded as a linearization operation, and each local RB is expected to better approximate the solutions for the corresponding subgroup. The following paragraphs will briefly describe several typical approaches for clustering the datasets and computing separate RBs.

The most widely used approach to cluster the pre-selected parameters or pre-computed snapshots is *k-means* algorithm. It is a widely used method for partitioning a dataset into k distinct clusters based on the similarity, e.g., *distance*, of data points. It consists of the following major steps. Firstly, k initial data points are randomly selected from the set. Then, the distance between each data point and all centroids is calculated, and each data point is assigned to the nearest centroid, forming k clusters. After that, the centroids are updated by calculating the mean of the data points in each cluster. The assignment and update steps are repeated iteratively until all data points are assigned.

An early implementation of the method for ROM is presented in [74, 133]. In this study, the snapshots are pre-computed for the entire parameter set, and then, the criterion for clustering is the distance between the high-fidelity solutions. If we have a centroid of i^{th} cluster, named $u_{c,i}$, the distance between $u_{c,i}$ and a snapshot u_j is calculated as

$$d_{i,j} = \|u_{c,i} - u_j\|_2 \,,$$

where we use u to denote the general field, $\|\cdot\|_2$ is the L^2 norm between the fields. Note that L^2 norm is applied in [74], but other metrics can also be used.

Washabaugh et al. [134] also used the k-means algorithm and L^2 norm between snapshots for clustering. The novelty of their work is that an overlap approach is applied to share the boundary snapshots between neighboring clusters, which can improve the accuracy of the ROM for the boundary parameters.

Pagani et al. [135] have compared the performance of clustering considering time-based, solution-based, and parameter-based criteria. They tried different clustering methods, one is the standard k-means, and the other is using the projection error[5] of the snapshots onto the local RB as the criterion for clustering.

We also recommend two recent publications about the usage of local RB [136, 137]. A significant difference between these two studies and the previous ones is that the clustering is performed in the parameter space instead of the solution space, indeed the distance (i.e., the L^2 norm) between parameters is calculated for clustering. It should be highlighted that, Chasapi et al. [136] adopted the *k-means variance*,

$$\mathcal{K} = \sum_{k=1}^{N_c} \sum_{\boldsymbol{\mu} \in \mathcal{P}_k} \|\boldsymbol{\mu} - \overline{\boldsymbol{\mu}}_k\|_2^2 \,,$$

as the criterion to choose the optimal number of clusters, where N_c is the number of clusters, $\mathcal{P}_k$ is the set of parameters in the k^{th} cluster, and $\overline{\boldsymbol{\mu}}_k$ is the centroid of the k^{th} cluster.

Regarding the k-means method, it is worth mentioning the work of Cai [138], in which they proposed an iterative procedure integrating k-means to optimize the clustering. The algorithm aims to maximize the distance between different clusters while minimizing the distance within the same cluster, and meanwhile, samples adjacent in the original space remain adjacent in the projected subspace. Although the method is not applied to ROM, we believe it is promising for the clustering of parameters or snapshots in ROM, and thus, it is worth further investigation.

Besides the k-means method, other techniques can also be applied to compute local RBs. Haasdonk el al. [139] proposed a grid based clustering method, in which the parameter space is partitioned into uniform partitions. They used an error estimator to test the accuracy of the local RB for each partition, and if the error is larger than a predefined threshold, the partition is further divided uniformly into smaller ones. Zou et al. [140] used a Voronoi tessellation to partition the parameter space. A Voronoi tessellation can be regarded as a simplified version of the k-means method, in which the centroids are fixed, and other data points are assigned to the nearest fixed point.

---

[5] Assume a set of orthonormal basis functions are stacked column-wise as Φ and a high-fidelity solution is stored as a vector $\mathbf{u}$, the projection error is defined as $\left\|\mathbf{u} - \Phi\Phi^T\mathbf{u}\right\|_2$.

3.3 Snapshots and Local Reduced Basis

Once the domain is properly partitioned and parameters are determined, the next step is to compute the local RB. The following paragraphs will briefly discuss three procedures for generating subdomain-level RBs.

Note that the concept of RB is defined for intrusive methodologies, whereas for non-intrusive methods, the low-dimensional representation is a standard expression. Nevertheless, to preserve the generality of the term, *RB* is still used as a common concept for both intrusive and non-intrusive frameworks.

3.3.1 Localized Global RB

In the first approach, parametric numerical solutions for the entire geometry are collected as a snapshot matrix, which is defined as

$$\underline{\underline{S}}_{N_h \times N_\mu} = \left[u_1, u_2, \ldots, u_{N_\mu}\right],$$

where N_μ is the total amount of parameters and N_h denotes the dimension of the high fidelity snapshot u_j. We use u_j to denote the general field for the j^{th} parameter, even though each column of $\underline{\underline{S}}$ is a vector of size N_h. The solution can be a scalar or vector field, and in this paper, we will use u to denote scalar fields or any general field, and $\mathbf{u}$ for vector fields.

u_j is defined as

$$u_j := u_j(\mathbf{x};\boldsymbol{\mu}_j)\text{for } j = 1, \ldots, N_\mu,$$

and $\boldsymbol{\mu}_j$ are either geometrical or physical parameters and $\mathbf{x}$ is the spatial coordinate.

Then, a reduction technique, e.g., POD, is adopted to compute dominant modes, which leads to the approximation:

$$u_j(\mathbf{x};\boldsymbol{\mu}_j) \approx \sum_{i=1}^{N_{\text{RB}}} a_i(\boldsymbol{\mu}_j) v_i(\mathbf{x}), \quad v_i \in \mathcal{V},$$

where N_{RB} is the number of modes, a_i are the coefficients, v_i are the basis functions defined in the entire domain Ω, $\mathcal{V}$ is the global reduced space.

Finally, the i^{th} localized function in subdomain Ω_m is defined as the restriction:

$$v_{m,i} = v_i|_{\Omega_m}.$$

Be aware that the localized functions $\{v_{m,i}\}_{i=1}^{N_{m,\text{RB}}}$ obtained by the restriction do not necessarily form a basis. For this reason, they are referred to as localized global basis functions rather than local basis functions. A comparison between the localized global RB and the local RB mentioned below is presented in [141].

3.3.2 Global Solutions and Local RB

The second technique also requires global FOM results that are decomposed before any further treatment. The subdomain-level reduced subspaces are computed using the local values. There are two procedures for computing the RBs.

- For the *individually partition* strategy (Sect. 3.1.3), we know $\Omega = \cup_{m=1}^{N_\Omega} \Omega_m$, so the local snapshots of Ω_m are obtained by the restriction

$$u_{m,j} := u(\mathbf{x};\boldsymbol{\mu}_j)|_{\Omega_m}\text{for } j = 1, \ldots, N_\mu,$$

  where $\mathbf{x}$ is the spatial coordinate and $u(\mathbf{x};\boldsymbol{\mu}_j)$ is the global solution for the j^{th} parameter, $\boldsymbol{\mu}_j$.
  Then, all parametric local fields of Ω_m are collected in a snapshot matrix

$$\underline{\underline{S}}_m = \left[u_{m,1}, u_{m,2}, \cdots, u_{m,N_\mu}\right].$$

  POD[6] can be applied to compute basis functions. Consequently, for Ω_m, the approximation with regard to $\boldsymbol{\mu}_j$ is

$$u_{m,j}(\mathbf{x};\boldsymbol{\mu}_j) \approx \sum_{i=1}^{N_{\text{RB},m}} a_{m,i}(\boldsymbol{\mu}_j) v_{m,i}(\mathbf{x}), \quad , v_{m,i} \in \mathcal{V}_m.$$

  An example of this strategy is shown in Fig. 14, in which a backward-facing step is decomposed into Ω_1 and Ω_2 [48]. Two different RBs are computed, namely, $\mathcal{V}_1$ and $\mathcal{V}_2$ for the two subdomains, respectively. The two RBs are *glued* for visualization, as presented in Fig. 14.
- In the *generic decomposition* approach, the domain is assembled by instantiating a few archetypes, which is expressed as $\Omega = \cup_{k=1}^{\hat{N}_\Omega} \cup_{m=1}^{N_k} \Omega_m^k$, where $\hat{N}_\Omega$ is the number of archetypes, N_k is the number of instantiations for archetype $\hat{\Omega}^k$, and Ω_m^k is the m^{th} instantiation of $\hat{\Omega}^k$ (See the details in Section 3.1.3).

  Similar to the previous strategy, the global solution $u(\mathbf{x};\boldsymbol{\mu}_j)$ is required ($\boldsymbol{\mu}_j$ denotes the j^{th} parameter for the entire problem). The local solution is obtained by restriction

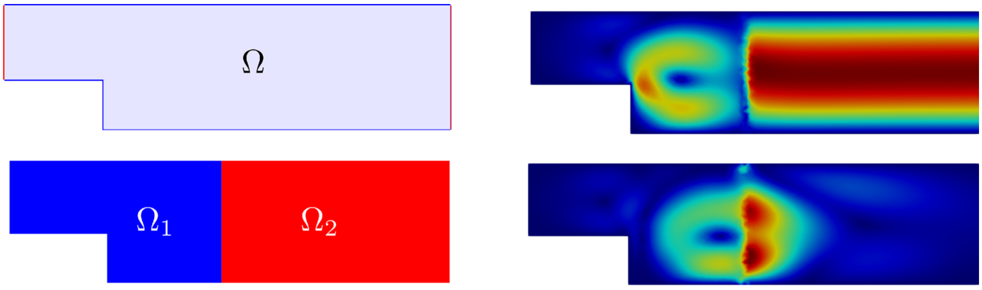
$$u_{m,j}^k := u\left(\mathbf{x};\boldsymbol{\mu}_j\right)\big|_{\Omega_m^k} \text{ for } j = 1, \ldots, N_\mu.$$

  Then, it is mapped to the reference block $\hat{\Omega}^k$ as

---

[6] POD is used here as an example, but other reduction techniques can also be applied.

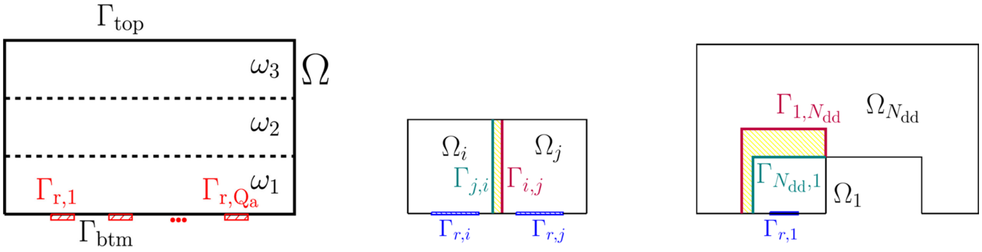
Fig. 14 Individual domain decomposition for a backward-facing step, $\Omega = \Omega_1 \cup \Omega_2$, and local RBs for a backward-facing step problem. The local velocity modes are *glued* together for visualization. Taken with permission from [58], copyright owned by the author



(a) Geometry and subdomains.

(b) Velocity modes.

Fig. 15 Domain decomposition of a thermo-hydro-mechanical system, consisting of many Ω^a_{int} and a single Ω^a_{ext}. Taken with permission from [25], copyright owned by the author



(a) Global model.

(b) Assembly of the geometry.

$$\hat{u}^k_{m,j} = \underline{T}^k(\boldsymbol{\xi}^k_m) \circ u^k_{m,j},$$

where $\underline{T}^k$ is the transformation that maps $\hat{\Omega}^k$ to Ω^k_m, and $\boldsymbol{\xi}^k_m$ is the set of geometrical parameters for Ω^k_m. Note that the transformation is applied to the spatial coordinate, so the solution is mapped to the reference block.

At this stage, the snapshot matrix for $\hat{\Omega}^k$ becomes

$$\underline{\underline{\hat{S}}}^k = \left[\underline{\underline{\hat{S}}}^k_1, \cdots, \underline{\underline{\hat{S}}}^k_{N_k}\right], \text{ and } \underline{\underline{\hat{S}}}^k_m = \left[\hat{u}^k_{m,1}, \cdots, \hat{u}^k_{m,N_\mu}\right].$$

POD or other techniques are applied to each $\underline{\underline{\hat{S}}}^k$ and obtain RBs. The basis vectors for $\hat{\Omega}^k$ are noted as

$$\hat{v}^k_i, \text{with } i = 1, \ldots, N^k_{\mathrm{RB},m}.$$

The local basis functions in Ω^k_m is obtained via the inverse transformation, $v^k_{m,i} = \underline{T}^{k^{-1}}(\boldsymbol{\xi}^k_m) \circ \hat{v}^k_i$, which can span a local reduced space $\mathcal{V}^k_m = \mathrm{span}\{v^k_{m,1}, \ldots, v^k_{m,N^k_{\mathrm{RB},m}}\}$. Thus, for Ω^k_m, the local solutions are approximated as

$$u^k_{m,j}(\mathbf{x}; \boldsymbol{\mu}_j) \approx \sum_{i=1}^{N^k_{\mathrm{RB},m}} a^k_{m,i}(\boldsymbol{\mu}_j) v^k_{m,i}(\mathbf{x}). \tag{2}$$

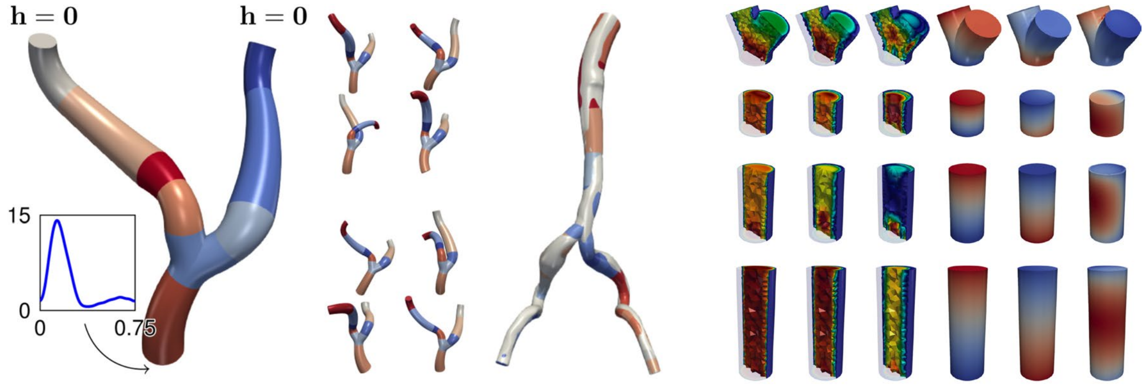
Figure 15 shows the decomposition of a thermo-hydro-mechanical system for radioactive waste disposal [25, 142]. One archetype components, Ω^a_{int}, is defined. Then, a set of instantiated Ω^a_{int} and one Ω^a_{ext} are assembled to build the model.

We also want to mention another scenario that uses solutions in partitions to compute the global RB. For FOM simulations running in parallel, the snapshots are stored in a distributed manner. Beattie et al. and Wang et al. [143, 144] proposed a method to compute the global RB without gathering the snapshots in the master process. The strategy is not relevant to the local RB, but it is worth mentioning for the sake of completeness in this review.

3.3.3 Localized Training and Oversampling

The third method is called *localized training*, which supports geometries assembled by reference parts. Instead of expensive high-fidelity modeling in the original large geometry, a set of parametric computations is carried out in several archetype components or much smaller systems containing archetype blocks. In this frame, it is assumed that the dynamics that occur in small-scale networks can represent the physical behavior of the original large-scale systems due to the existence of periodicity and self-similarity in the geometry and physics. The approach can significantly accelerate the most expensive part for building a ROM, i.e., the offline stage, which has the potential to make ROMs feasible for real-world applications.

The main difference between this strategy and the previous one is the way to parameterize the problem and collect snapshots. Figure 16 illustrates this approach applied to blood vessel flow [33]. The model is built from four archetypes, one branch and three straight pipes (Fig. 16b). High resolution simulations are performed on a small-scale geometry that contains all archetypes (Fig. 16a left). Parameterization is achieved locally to the small geometry and to



(a) FOM geometry. (b) Modes in archetypes.

Fig. 16 The illustration of the localized training strategy for a blood vessel. (**a**) The high-fidelity computational domain (left), geometrical parameterized samples (middle), and an assembled large domain (right). (**b**) Subdomain-level velocity and pressure modes for each archetype. Figures taken with permission from [33], copyright owned by Elsevier

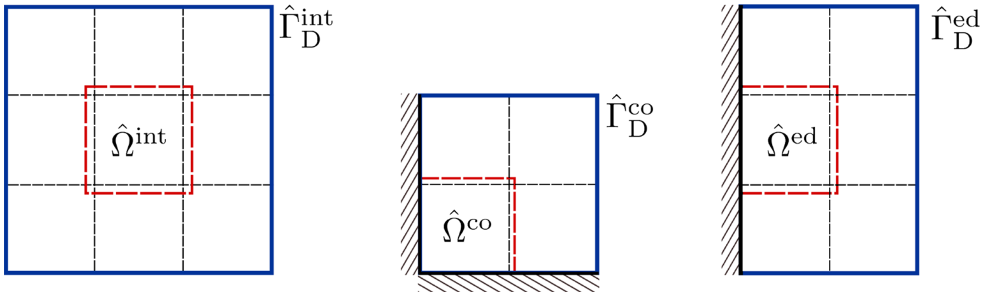


Fig. 17 A sketch of the *oversampling* strategy for three archetype components, i.e., $\hat{\Omega}^{\mathrm{int}}$, $\hat{\Omega}^{\mathrm{co}}$ and $\hat{\Omega}^{\mathrm{ed}}$. Three small-scale models are constructed to generate high-fidelity solutions of each generic subdomain. Redrawn based on [146]

each archetype (Fig. 16a middle). Subdomain-level snapshots are then collected to compute local reduced bases (see Fig. 16b). At the online stage, archetype blocks are assembled into the full geometry (Fig. 16b right), and the local reduced bases are mapped to each instantiated partition to compute the global solution.

Let us clarify the notation for the strategy. Assume the k^{th} archetype $\hat{\Omega}^k$ is parameterized by $\left\{\hat{\boldsymbol{\mu}}_j^k, j = 1, \ldots, \hat{N}_\mu^k\right\}$, where $\hat{N}_\mu^k$ is the number of parameters for $\hat{\Omega}^k$. The corresponding local solutions are defined as

$$\hat{u}^k(\hat{\mathbf{x}}; \hat{\boldsymbol{\mu}}_j^k) \text{for } j = 1, \ldots, \hat{N}_\mu^k,$$

where $\hat{\mathbf{x}}$ is the spatial coordinate in the reference block $\hat{\Omega}^k$.

The archetype-level snapshot matrix is defined as

$$\underline{\underline{\hat{S}^k}} = \left[\hat{u}_1^k, \cdots, \hat{u}_{\hat{N}_\mu^k}^k\right],$$

where $\hat{u}_j^k := \hat{u}^k(\hat{\mathbf{x}}; \hat{\boldsymbol{\mu}}_j^k)$. Then, a reduction technique is applied to compute the basis functions

$$\hat{v}_i^k, \text{with } i = 1, \ldots, N_{\mathrm{RB},m}^k.$$

At the online stage, the computational domain is assembled by instantiating the archetypes. So the basis functions are mapped to the instantiated blocks via the geometrical transformation, which is expressed as $v_{m,i}^k = \underline{T}^{k^{-1}}(\boldsymbol{\xi}_m^k) \circ \hat{v}_i^k$. The local solution in Ω_m^k is approximated as

$$u_{m,j}^k(\mathbf{x}; \boldsymbol{\mu}_j) \approx \sum_{i=1}^{N_{\mathrm{RB},m}^k} a_{m,i}^k(\boldsymbol{\mu}_j) v_{m,i}^k(\mathbf{x}),$$

where $\boldsymbol{\mu}_j$ denotes the j^{th} parameter for the entire problem.

In order to compute modes that are representative of the physical phenomena appearing in the global solution, an *oversampling* strategy has been proposed [145]. The FOM simulations are performed in geometries that are slightly larger and comprise several archetypes. To the extended domains, some parameterized boundary conditions are applied, as shown in Fig. 17. The solutions within the reference subdomains are extracted by restriction to compute corresponding RBs. The oversampling approaches can identify low-dimensional structures and have been suggested and used extensively [50, 146–148]. Besides, the increasing computation cost for FOM simulations remains acceptable.

3.3.4 Reduced Basis for Interior, Interface, and Boundary.

We finalize this section by discussing the practical formation of the snapshot's matrix regarding the interior Ω, interfaces

Γ (or the local boundary $\partial\Omega_m$), and global boundary $\partial\Omega$. Recall that the matrix will be treated with a reduction algorithm to obtain the RBs.

Fig. 18 illustrates the three strategies to process offline solutions: (i) solutions of the three zones -interior, interfaces, and boundaries- are gathered, and a single compact reduced space is constructed; (ii) processing the interior and the interface together and the boundary separately; or (iii) each zone is post-processed separately.

The procedure to follow depends on the partition strategies described above and the coupling algorithm introduced in the following sections.

3.4 Acceleration Techniques for the Offline and Online Stages

As we mentioned before, the local ROM aims to reduce the computational cost of high-fidelity simulations to enhance the feasibility of ROMs for real-world applications. Additionally, we will briefly introduce two approaches that can further accelerate the offline and online stages of ROMs, namely *incremental ROM* and the *hyper-reduction*. The two strategies are integrated into ROMs in the process of generating snapshots and computing the RBs, and are thus described here. Be aware that they can be applied to both intrusive and non-intrusive ROMs, and also to both global and local ROMs. Due to the flexibility, they are employed together in many studies. Their application for local ROMs will be indicated in Sections 4 and 5.

3.4.1 Incremental Reduced Order Model

In Sect. 3.2.1, we have described the adaptive sampling strategy to optimally select parameters for the offline stage. The same principle can be applied to the online stage as well, which is called *online basis enrichment*. ROMs created in this manner are also known as *incremental ROMs*.

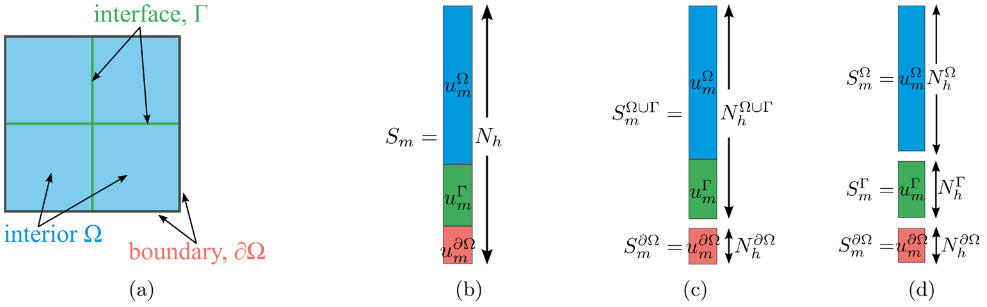
Similar to adaptive sampling, an *error indicator* is used to evaluate the accuracy of the ROM with respect to the current parameter or time step. If the estimated error during the online stage is larger than a predefined threshold, a high-fidelity simulation is performed to enrich the reduced basis. Consequently, the incremental ROM can be regarded as a mixture of offline and online stages. ROM is applied for approximation, and meanwhile, the FOM simulation is performed to enrich the reduced basis when necessary.

Due to the variety of error indicators and enrichment strategies, it is difficult to give a comprehensive review of the incremental ROMs. Thus, we will only roughly list the observed publications and classify them into intrusive and non-intrusive groups.

Intrusive incremental ROMs are studied more extensively, and they have been applied to various problems, such as steady-state and time-dependent heat equation [149, 150], elastodynamics and poroelasticity [150], Burgers' equation [151–153], Euler equations [154, 155], advection-diffusion-reaction phenomena [156], transport-dominated problems [157], Navier-Stokes equations [158], structural mechanics [154, 159], and FitzHugh-Nagumo equations [156, 160]. The integration of incremental ROMs in non-intrusive frameworks is also gaining attention, see more in [161, 162], especially the integration of active learning in [91, 163].

Note that when snapshots are generated adaptively, it is not necessary to recompute the RB from the full snapshot matrix whenever new data become available. Instead, the RB can be updated incrementally by means of *incremental POD* or *incremental Singular Value Decomposition (SVD)* [149, 150, 164, 165], in which only the new snapshots are incorporated into the update. This strategy can substantially reduce both memory requirements and computational cost for enriching the RB.

Moreover, for large-scale CFD problems, data should be stored in a spatially distributed manner due to domain decomposition and MPI-parallel solvers. Kühl et al. and Costanzo et al. [166, 167] have proposed a parallel incremental SVD algorithm, which can update the RB directly for large-scale spatially parallel & distributed but temporally serial data. The methods were demonstrated for several fluid-dynamics applications, including laminar flow around a circular cylinder, a submerged hydrofoil in turbulent two-phase flow, a Kriso container ship hull in harmonic head waves [166], and mesh-free smooth particle hydrodynamics simulations such as sloshing, an impinging jet, and a Pelton turbine runner [167]. We believe the incremental SVD-based approaches is particularly valuable for the implementation



(a) (b) (c) (d)

Fig. 18 Snapshots structures. (**a**) a domain and three regions: interior, Ω; interface, Γ; and global boundary, $\partial\Omega$. (**b**) a compact subdomain snapshot. (**c**) Subdomain and boundary snapshots. (**d**) Separated snapshots of ROMs for unsteady and large-scale simulations in industrial applications, and it is an ongoing research focus in the ROM community.

3.4.2 Hyper-Reduction Technique

Hyper-reduction is employed to accelerate the online stage of ROMs containing *nonlinear terms*, which is also known as *non-affine operators*. We will briefly explain why the evaluation of nonlinear terms is expensive and how hyper-reduction can help to accelerate the online stage.

Assume we have a nonlinear term $\underline{\mathcal{N}}(u;\boldsymbol{\mu})$ in the governing equations parameterized by $\boldsymbol{\mu}$, and the solution is approximated as $u \approx \sum_{i=1}^{N_{\mathrm{RB}}} a_i(\boldsymbol{\mu}) v_i$, where v_i are the basis functions and a_i is the corresponding coefficients. The projected nonlinear term is expressed as

$$\underline{\underline{N_r}} = \underline{\underline{V}}^T \underline{\underline{N}},$$

where $\underline{\underline{N_r}}$ is the reduced nonlinear matrix, $\underline{\underline{V}}$ is the matrix of basis functions, and $\underline{\underline{N}}$ is the high-fidelity nonlinear matrix, whose elements are computed by $\underline{\mathcal{N}}\left(\sum_{i=1}^{N_{\mathrm{RB}}} a_i(\boldsymbol{\mu}) v_i\right)$.

Be aware that the evaluation of $\underline{\mathcal{N}}\left(\sum_{i=1}^{N_{\mathrm{RB}}} a_i(\boldsymbol{\mu}) v_i\right)$ requires the computational cost proportional to the dimension of the high-fidelity model. Moreover, since $\underline{\mathcal{N}}$ is a nonlinear operator, it should be recomputed for each new approximation $u \approx \sum_{i=1}^{N_{\mathrm{RB}}} a_i(\boldsymbol{\mu}) v_i$. Although the basis functions v_i are pre-computed in the offline stage, the coefficients a_i depending on the parameter $\boldsymbol{\mu}$ are estimated during the online stage. Consequently, the nonlinear term $\underline{\mathcal{N}}\left(\sum_{i=1}^{N_{\mathrm{RB}}} a_i(\boldsymbol{\mu}) v_i\right)$ and the projected matrix $\underline{\underline{N_r}}$ have to be calculated at each time step or for each parameter during the online stage, which is very expensive.

We may conclude that the cost for evaluating $\underline{\mathcal{N}}\left(\sum_{i=1}^{N_{\mathrm{RB}}} a_i(\boldsymbol{\mu}) v_i\right)$ is due to the high dimension of the FOM. Hyper-reduction aims to evaluate the nonlinear term over a small subset of discretization points (or elements), instead of the entire domain, and hence, the computational cost is significantly reduced.

The selection of the subset of points and the approximation of the nonlinear term are the two main steps for hyper-reduction. The simplest way to determine the subset of points is to randomly choose a few points from the entire domain, which is known as the *collocation method*. However, the collocation method is not optimal and can lead to large errors. Consequently, four more sophisticated hyper-reduction techniques have been proposed in the past decades. These are *Empirical Interpolation Method* (EIM) [168], *Discrete Empirical Interpolation Method* (DEIM) [169], *Gappy POD* [170], and *Gauss-Newton with Approximated Tensors* (GNAT) [171].

The four methods differ mainly in the way to select the sampling points and approximate the nonlinear term. EIM is the first hyper-reduction approach proposed in the literature, and DEIM is a discrete variant of EIM, which is more practical for numerical implementation with discretized computation domains and governing equations. Gappy POD is a method that optimally selects the sampling points for evaluating the nonlinear term by solving a least-squares problem. GNAT is a more recent method that is initially designed for structural mechanics and fluid dynamics, respectively.

The collocation method is adopted in [19, 54, 172]. EIM and DEIM are the most widely used techniques; for more details see [34, 136, 173, 174]. Details on Gappy POD appear in [54, 172], and GNAT is described in [171]. We also recommend the study by Hoang et al. [175], which compares several hyper-reduction techniques for a domain decomposition problem. Moreover, other methods that can be regarded as hyper-reduction techniques include the empirical quadrature procedure [176] and the empirical cubature method [177].

It is worth mentioning a series of recent publications by Bai et al. [178–181], in which intrusive incremental ROM and hyper-reduction (DEIM and Gappy POD) are combined to model the heat equation and nonlinear diffusion-reaction systems. Their results indicate that hyper-reduction can reduce the computational cost of the online stage by approximately 50%.

Research into both incremental ROM and hyper-reduction is very active, and various improvements have been proposed to enhance accuracy, stability, and efficiency. They are widely used to build global ROMs, and we believe they are also promising for local ROMs due to their flexibility and efficiency. However, the application of these techniques for local ROMs is still limited, and more research is needed to explore their potential for local ROMs. A more detailed discussion of these topics and their mathematics is beyond the scope of this review. We recommend the review [182] for hyper-reduction techniques, while for incremental ROMs, see the papers cited above, especially those by Bai et al.

4 Projection-Based Coupling Algorithms

Galerkin projection is the fundamental procedure utilized to derive the weak formulation of PDEs. It is widely adopted to formulate and solve high-fidelity problems in the frame of many discretization algorithms, such as the spectral method [183] and *Finite Element Method* (FEM) [5].

The technique is also commonly employed to derive intrusive reduced systems. Notably, it is utilized in our domain of interest: ROMs with domain decomposition. However, the usage of this approach is challenging. For intrusive local ROMs, the original PDEs have to be manipulated. In addition, the contributions of the interfaces and interactions between subdomains should also be included in the final formulations. In summary, intrusive ROMs assembling multiple partitions are effective but complex in terms of numerics and programming. Therefore, in this section, we provide a brief introduction to the various formulations of projection-based coupling algorithms, including both monolithic and iterative procedures. Fig. 19 displays the categories of the two groups.

The differences and advantages of the two frameworks can be briefly summarized below. The monolithic methodology is more intrusive than the latter. It aims to create a large ROM involving all subdomains. Due to the compact reduced system, it is more favorable in terms of stability, convergence, and precision compared to iterative approaches [19]. However, the higher level of intrusiveness also results in extensive numerical and programming complexities. As shown in Fig. 19, the monolithic category is more diverse.

In contrast, the iterative procedure aims to create multiple separate sub-problems corresponding to all partitions. The local ROMs are solved iteratively with updated boundary conditions from neighbors. The interface jump or error decreases iteratively, and the solver stops when it satisfies the predefined convergence criterion.

In the following sections, we will first give the basic formulation, namely, the strong and weak forms for PDEs. Then, the coupling techniques are explained. Note that a priori and a posteriori error analyses are usually included in the investigations, which are not illustrated in the review for brevity.

4.1 Strong and Weak Form

We first present the general formulation, which will serve as the foundation for the methods discussed in the following sections. The derivations of weak formulations from the strong form of a PDE have been extensively documented in the literature [5, 53, 183]. In this work, we will consider the strong form of a generic parametric PDE for the unknown field $u\left(\mathbf{x}, t; \boldsymbol{\mu}_j\right)$,

$$\dot{u}\left(\mathbf{x}, t; \boldsymbol{\mu}_j\right) + \mathcal{Q}\left(u\left(\mathbf{x}, t; \boldsymbol{\mu}_j\right); \boldsymbol{\mu}_j\right) = f\left(\mathbf{x}, t; \boldsymbol{\mu}_j\right), \qquad \text{in } \Omega \times [0, t], \tag{3}$$

where $\dot{u}$ is the time derivative of u, $\mathcal{Q}$ is a generic operator, which can be linear or nonlinear. f is a source term. $\boldsymbol{\mu}_j$ is the j^{th} parameter for the entire problem, which can be geometrical, physical, or other types of parameters. $\mathbf{x}$ and t denote spatial and temporal coordinates, respectively. Here u can be regarded as a general field. We will use $\mathbf{u}$ to denote a vector of solution variables if necessary.

The boundary conditions can be expressed as

$$\begin{aligned} u(\mathbf{x}, t; \boldsymbol{\mu}_j) &= g_D\left(\mathbf{x}, t; \boldsymbol{\mu}_j\right), &\quad \text{in } \Gamma_D \times [0, t], \\ \nabla u(\mathbf{x}, t; \boldsymbol{\mu}_j) \cdot \mathbf{n} &= \mathbf{g}_N\left(\mathbf{x}, t; \boldsymbol{\mu}_j\right), &\quad \text{in } \Gamma_N \times [0, t], \end{aligned}$$

where $\mathbf{n}$ is the normal vector on the boundary. g_D and $\mathbf{g}_N$ are the assigned values for Dirichlet and Neumann conditions, respectively.

The initial condition is given by

$$u(\mathbf{x}, 0; \boldsymbol{\mu}_j) = u_0(\mathbf{x}; \boldsymbol{\mu}_j), \qquad \text{in } \Omega,$$

where u_0 is the initial value of the solution.

To simplify the notation, hereafter, we omit the dependent variables $(\mathbf{x}, t)$, e.g., $\dot{u}(\boldsymbol{\mu})$, and $\mathcal{Q}\left(u; \boldsymbol{\mu}_j\right)$. The weak formulation is derived by means of the inner product with test functions $w \in \mathcal{W}$. The equations are required to hold only with respect to a certain test space $\mathcal{W}$, whose exact definition will vary among the formulations and will be pertinently detailed. Applying the integral-by-parts formula,

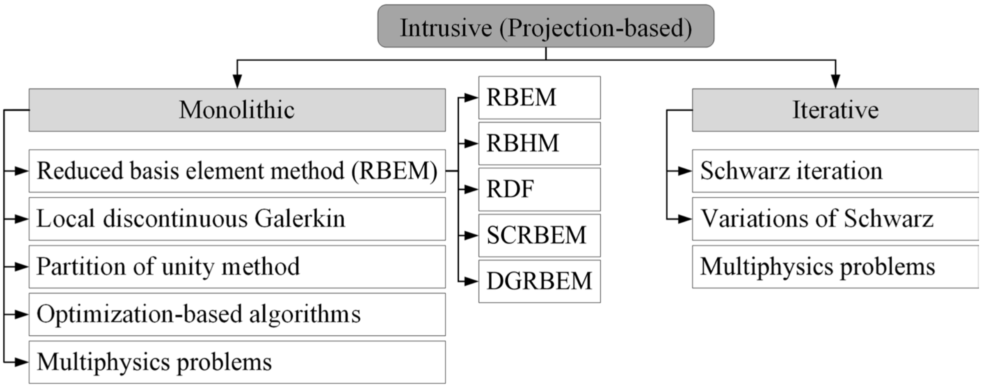


Fig. 19 The categorization of intrusive techniques

volumetric integrals are divided into volumetric and surface parts,

$$\underline{\mathcal{Q}}(u;\boldsymbol{\mu}_j) = \underline{\mathcal{Q}}_\Omega(u;\boldsymbol{\mu}_j) + \underline{\mathcal{Q}}_\Gamma(u;\boldsymbol{\mu}_j),$$

where $\underline{\mathcal{Q}}_\Omega$ and $\underline{\mathcal{Q}}_\Gamma$ denote the volumetric and surface contributions, respectively.

Then, by applying inner product with test function w, we rewrite Eq. 3 as the *weak form*,

$$\left(\dot{u}, w;\boldsymbol{\mu}_j\right) + \underline{\mathcal{Q}}\left(u, w;\boldsymbol{\mu}_j\right) = \underline{\mathcal{F}}(w;\boldsymbol{\mu}_j), \quad \text{in } \Omega \times [0,t]\,,$$

or

$$(\dot{u}, w;\boldsymbol{\mu}_j) + \underline{\mathcal{Q}}_\Omega\left(u, w;\boldsymbol{\mu}_j\right) + \underline{\mathcal{Q}}_\Gamma\left(u, w;\boldsymbol{\mu}_j\right) = \underline{\mathcal{F}}(w;\boldsymbol{\mu}_j), \quad \text{in } \Omega \times [0,t]\,,$$

where $(\cdot,\cdot)$ denotes the inner product operation. The parameterized forms are defined as operators acting on the spaces of test and trial functions, such that $\underline{\mathcal{Q}}_\Omega(u,w;\boldsymbol{\mu}_j) = \left(\underline{\mathcal{Q}}_\Omega(u;\boldsymbol{\mu}_j), w\right)$, $\underline{\mathcal{Q}}_\Gamma(u,w;\boldsymbol{\mu}_j) = \left(\underline{\mathcal{Q}}_\Gamma(u;\boldsymbol{\mu}_j), w\right)$, $\underline{\mathcal{F}}(w;\boldsymbol{\mu}_j) = \left(f(\boldsymbol{\mu}_j), w\right)\big|_{\overline{\Omega}}$. Remark that the global boundary contributions at Γ_D and Γ_N are involved in $\underline{\mathcal{F}}(w;\boldsymbol{\mu}_j)$ for simplification.

We may approximate u as a function of trial functions $v_j \in \mathcal{V}$, with $\mathcal{V}$ a generic trial functional space. We get

$$u(\mathbf{x}, t;\boldsymbol{\mu}_j) \approx \sum_{j=1}^{n} a_j(t;\boldsymbol{\mu}_j) v_j(\mathbf{x}),$$

where n is the dimension of both $\mathcal{V}$ and $\mathcal{W}$ and $a_j(t;\boldsymbol{\mu}_j)$ are the weighted coefficients of the basis vectors. To obtain a system containing n equations we use inner product with test functions, $w_i \in \mathcal{W},\ i = 1, 2, \cdots, n$,

$$\begin{aligned}&\left(\sum_{j=1}^{n} \dot{a}_j v_j, w_i;\boldsymbol{\mu}_j\right) + \underline{\mathcal{Q}}_\Omega\left(\sum_{j=1}^{n} a_j v_j, w_i;\boldsymbol{\mu}_j\right)\\ &+ \underline{\mathcal{Q}}_\Gamma\left(\sum_{j=1}^{n} a_j v_j, w_i;\boldsymbol{\mu}_j\right) = \underline{\mathcal{F}}(w_i;\boldsymbol{\mu}_j) \quad v_j \in \mathcal{V}, w_i \in \mathcal{W}.\end{aligned} \tag{4}$$

The unknowns of the system are the coefficients a_j and their time derivatives are $\dot{a}_j$. The DoFs of the system depend on the dimension of the test and trial spaces, indicated as n. Note that the DoFs of the original high-fidelity system are related to the number of cells/elements, denoted by N_h. The reduction is achieved by choosing $n \ll N_h$. The choice of the test and trial spaces is crucial for the accuracy and stability of the reduced system.

The algebraic form of the ROM 4 is

$$\underline{\underline{M}}\dot{\mathbf{a}} + \underline{\underline{C}}\mathbf{a} + \underline{\underline{D}}\mathbf{a} = \underline{\underline{F}}, \tag{5}$$

where $\mathbf{a} = [a_1, a_2, \ldots, a_n]^T$ is a vector containing coefficients of modes. $\dot{\mathbf{a}}$ is the corresponding time derivative. The element of the matrices are addressed as

$$\begin{aligned}M_{ij} = (v_j, w_i;\mu_j), \quad C_{ij} = \underline{\mathcal{Q}}_\Omega(v_j, w_i;\mu_j), \quad D_{ij}\\ = \underline{\mathcal{Q}}_\Gamma(v_j, w_i;\mu_j), \quad F_i = \underline{\mathcal{F}}(w_i;\mu_j),\end{aligned}$$

where i and j are the indices for rows and columns, respectively.

In case the test and trial spaces are the same, the projection is called *Bubnov-Galerkin* [184], which is also referred to as *standard Galerkin*. When the two sets of basis functions are different, it leads to the *Petrov-Galerkin* or *non-standard Galerkin* [5]. We kindly refer the reader to the books [12, 14] for detailed instructions about the Galerkin projection-based ROM.

We will now start reviewing different intrusive methods available for the coupling task.

4.2 Monolithic

We begin our review from the first category of methods, known as *monolithic* techniques. These approaches rely on a set of local RBs, either a generic basis or subdomain-specific ones, to construct the global solutions. Interface contributions from adjacent partitions are incorporated into the original formulation either as additional terms or through a new set of equations enforcing the constraints. A noticeable advantage of these methods over Global ROMs is that each subdomain is governed by a smaller number of parameters, whether geometrical or physical, allowing for greater flexibility in parameterization. Moreover, for localized sensitivity and uncertainty analysis, the offline training effort is significantly reduced. Note that coupled systems for multiphysics are generally different from a single-physics problem, so they are described in a separate section.

4.2.1 Reduced Basis Element Method

Reduced Basis Element Method (RBEM) was created by Maday and Rønquist [44] in 2002, based on the *mortar element method* [185, 186] that was already available and widely utilized for the decomposition of domains in high-resolution problems. As one of its main characteristics, the algorithm employs *Lagrange multipliers* to enforce continuity across partition interfaces. As a result, the reduced system comprises (i) the original PDEs containing only volumetric terms, source, and Boundary Conditions (BCs); (ii) additional interface-integral terms corresponding to

Lagrange multipliers; (iii) a new set of equations to constrain the jumps between adjacent subdomains [44].

The decomposition into archetypes (see Sect. 3.1.3) is utilized by RBEM to generate subdomains. We recall that, for an instantiated component Ω_m^k, the local data $u_m^k(\boldsymbol{\mu})$ can be represented as a linear combination of local basis functions v_m^k. At this stage, and for simplicity, we combine the indices of the archetype k and the instantiation m into a single one, namely $\Omega = \cup_{m=1}^{N_\Omega} \Omega_m$. This notation results in the local value

$$u_m(\boldsymbol{\mu}) \approx \sum_{i=1}^{N_{\mathrm{RB},m}} a_{m,i}(\boldsymbol{\mu}) v_{m,i}, \tag{6}$$

where $N_{\mathrm{RB},m}$ is the number of RBs for Ω_m, and $a_{m,i}(\boldsymbol{\mu})$ is the coefficient of the i^{th} RB for Ω_m. The basis functions are generated by the localized training strategy, which has already been detailed in Sect. 3.3.3.

Now, we turn to a multiple-partition system and select one subdomain Ω_m as an example to explain the construction of a ROM based on RBEM. The interface connected to Ω_m and its neighbor Ω_n is denoted by $\Gamma_{[mn]} := \partial\Omega_m \cap \partial\Omega_n$[7]. As Ω_m might have several adjacent partitions, the indices of all neighbors are included in a set $N_\Gamma(m) = \{n | \partial\Omega_m \cap \partial\Omega_n \neq \emptyset\}$.

Before illustrating the governing equations, we introduce notations for clarification. We start with the Hilbert space in $\overline{\Omega}$, $H^1(\overline{\Omega})$[8]. The test space belongs to $H^1(\overline{\Omega})$ and its trace is equal to g_D on Γ_D is $\mathcal{W}_{g_D} \subset H^1_{g_D,\Gamma_D}(\overline{\Omega})$. Thus, for Ω_m , $\mathcal{W}_{m,0} \subset H^1_{0,\Gamma_D}(\overline{\Omega}_m)$ denotes functions in the test space that are homogeneous on the local $\Gamma_D \in \partial\Omega_m$. The trial space is $\mathcal{V}_m \subset H^1(\overline{\Omega}_m)$.

In $\Gamma_{[mn]}$, we also define the functional basis space $\mathbb{L}_{[mn]}$ [9] and the basis vectors $\zeta_{[mn],i} \in \mathbb{L}_{[mn]}$. A possible choice of $\mathbb{L}_{[mn]}$ is as the space generated by a set of orthogonal polynomials on the interface $\Gamma_{[mn]}$, e.g., the Chebyshev polynomials. We refer interested readers to the references [22, 33] for details. The Lagrange multiplier defined on $\Gamma_{[mn]}$ can be expressed in terms of the basis in the interface as

$$\lambda_{[mn]} \approx \sum_{i=1}^{N_{\lambda,[mn]}} q_{[mn],i} \zeta_{[mn],i},$$

with $q_{[mn],i} \in \mathbb{R}$and $\zeta_{[mn],i} \in \mathbb{L}_{[mn]}$,

where $q_{[mn],i}$ is the coefficient of the i^{th} basis vector in $\mathbb{L}_{[mn]}$, and $N_{\lambda,[mn]}$ is the number of basis vectors adopted for the approximation.

The generic RBEM governing equation for Ω_m and its interfaces is written as

$$\begin{aligned}&(\dot{u}, w_{m,i}) + \underline{\mathcal{Q}}_\Omega(u, w_{m,i})\\ &+ \sum_{n\in N_\Gamma(m)} \int_{\Gamma_{[mn]}} \lambda_{[mn]} w_{m,i} = \underline{\mathcal{F}}(w_{m,i}), \quad w_{m,i} \in \mathcal{W}_{m,0},\end{aligned} \tag{7}$$

where parameters $\boldsymbol{\mu}_j$ are omitted for brevity, the global boundary terms are involved in $\underline{\mathcal{F}}(w_{m,i})$, and not explicitly shown in the equation for simplification. Besides, since the test functions $w_{m,i}$ are homogeneous on $\Gamma_{[mn]}$, $\underline{\mathcal{Q}}_\Gamma$ is eliminated from the equation.

For every interface, $\lambda_{[mn]} = -\lambda_{[nm]}$, and the jump between adjacent subdomains is constrained by the following equation,

$$\int_{\Gamma_{[mn]}} \zeta_{[mn],i}(u_m - u_n) = 0 \quad \forall \zeta_{[mn],i} \in \mathbb{L}_{[mn]}. \tag{8}$$

Note that the global Dirichlet boundary conditions are weakly imposed by the Lagrange multipliers in Eq. 8.

The RBEM system is finally built based on Eqs. 7 and 8 for all subdomains and interfaces. The unknowns of the system are the coefficients of the RBs for all subdomains and the coefficients of the Lagrange multipliers for all interfaces. We recommend readers to refer to Appendix A.1 for a detailed derivation of the RBEM formulation, where we show the construction of the local reduced systems and the assembly of the global system.

To the best of our knowledge, the RBEM was the first technique developed to combine domain decomposition and ROMs. Maday and Rønquist proposed this approach in [44], applying it to the Laplacian equation for a geometrically parameterized thermal fin problem (Fig. 7a) [106]. Subsequent work with Lovgren extended RBEM to handle the steady Stokes and Navier-Stokes equations in a 2D blood vessel geometry [22, 187].

Further developments of RBEM have since emerged. Chen et al. [188] adapted the technique for the time-harmonic Maxwell's equation. A recent contribution utilizing RBEM to the unsteady Navier-Stokes flow in a 3D blood vessel can be found in [33]. We present their model in Sect. 3.3.3 as an example for generic spatial decomposition (see Fig. 7b) and RBs for subdomains (see Fig. 16).

Besides its incorporation in RBEM, the Lagrange multipliers can *glue* different interfaces. Thus, the aforementioned formulation can be used to assemble several arbitrary geometries without algorithmic modifications. Farhat et al. followed this ideology to analyze the Helmholtz problems

---

[7] The subscript $[mn]$ is a combined index to designate a single face.

[8] Note that $\overline{\Omega} = \Omega \cup \partial\Omega$.

[9] $\mathbb{L}_{[mn]} \subset H_{00}^{-1/2}(\Gamma_{[mn]})$ is a type of Hilbert space defined on the interface $\Gamma_{[mn]}$, see [53] for further details.

with plane waves in multiple subdomains [189]. Recent studies by Tezaur et al. [190, 191] have further demonstrated the method's capability. They managed to achieve FOM-ROM and ROM-ROM systems of an advection-diffusion equation. Note that a FOM-ROM system contains two subdomains, and they are simulated by a high-fidelity FOM and a ROM, respectively. Additionally, the ideology can be extended to a coupling of multiple FOMs and ROMs.

4.2.1.1 Reduced Basis Hybrid Method Based on RBEM, Iapichino et al. [192] proposed a modification, the so-called *Reduced Basis Hybrid Method* (RBHM). They intended to parametrize a steady Stokes problem for a blood vessel. The approach is almost the same as the RBEM except for the construction of RBs. To ensure continuity and consistency of the normal stress during the online stage, they used *hybrid* RBs to approximate velocity fields, which contain three components: (i) high-fidelity solutions collected from training models, $\hat{\mathbf{u}}^k(\boldsymbol{\mu}_j)$[10]; (ii) coarse mesh solutions $\mathbf{u}^{\mathrm{H}}(\boldsymbol{\mu}_p)$ of the whole problem considering parameters $\boldsymbol{\mu}_p$ that need to be predicted with the ROM, and (iii) velocity supremizer basis vectors $\mathrm{s}_m^{\mathrm{H}}(\boldsymbol{\mu}_p)$. The last one entitles the basis to fulfill the inf-sup condition [12], enhancing the numerical stability (details on *supremizer enrichment* technique are provided by references [14, 193, 194]). Consequently, in each subdomain Ω_m, the velocity is expressed as

$$\begin{aligned}\mathbf{u}_m(\boldsymbol{\mu}_p) &\approx \underline{T}^k(\boldsymbol{\xi}_m^k) \circ \left(\sum_{i=1}^{N_\mu} a_{m,i}\hat{\mathbf{u}}^k(\boldsymbol{\mu}_i) + \sum_{i=1}^{N_\mu} c_{m,i}\hat{\mathbf{s}}^k(\boldsymbol{\mu}_i)\right) + b\mathbf{u}_m^{\mathrm{H}}(\boldsymbol{\mu}_p) + d\mathbf{s}_m^{\mathrm{H}}(\boldsymbol{\mu}_p)\\ &= \sum_{i=1}^{N_\mu} a_{m,i}\mathbf{v}_{m,i} + b\mathbf{u}_m^{\mathrm{H}}(\boldsymbol{\mu}_p) + \sum_{i=1}^{N_\mu} c_{m,i}\mathbf{s}_{m,i} + d\mathbf{s}_m^{\mathrm{H}}(\boldsymbol{\mu}_p),\end{aligned}$$

where $\underline{T}^k$ is the transformation from the reference shape to the instantiated one (see 1), and $a_{m,i}$, b, $c_{m,i}$, and d are the coefficients of the basis vectors. Then, the RBHM formulation is derived by substituting the above expression into the RBEM formulation.

4.2.2 Reduced Basis, Domain Decomposition, and Finite Element Method

The *Reduced basis, Domain decomposition, and Finite element* method (RDF) procedure is named from the combination of the three approaches. It is designed for problems considering a generic decomposition with non-overlapping subdomains [59, 195].

The main difference with respect to RBEM and RBHM is that RDF does not need Lagrange multipliers that impose interface equality. Instead, extra interface basis functions are enriched to the solution space and ensure the continuity across interfaces. The construction of RBs is explained as follows.

The *internal* RDF RBs are computed similarly to RBEM, using solutions of sub-problems defined in reference shapes. The sub-models are parameterized with random combinations *Fourier series* or *Lagrange polynomials* as Dirichlet BCs. Lifting functions are applied to obtain basis vectors vanishing on Dirichlet boundaries. The lifted results are employed as RBs, which satisfy

$$\mathcal{V}_{m,0} = \mathcal{W}_{m,0} \subset H^1_{0,\Gamma_D}(\overline{\Omega}_m). \tag{9}$$

The authors demonstrated that the additional local boundary parameterization allows RDF to represent complex interface profiles better and thus capture variations of the final global solution more accurately.

The second type of RBs are the *interface* ones. They are constructed by using the FEM shape functions ϕ defined on the interfaces. Assume a reference internal interface $\hat{\Gamma}$ with $\hat{N}^h_{\Gamma}$ nodes, the basis space defined on $\hat{\Gamma}$ is

$$\Phi^{\hat{\Gamma}} := \operatorname{span}\left\{\phi_l^{\hat{\Gamma}} \in \mathcal{V}^h\big|_{\hat{\Gamma}},\ l = 1, \cdots, N^h_{\hat{\Gamma}}\right\},$$

where $\phi_l^{\hat{\Gamma}}$ are Lagrange basis functions for the *finite element* nodes and $\mathcal{V}^h$ is the high dimensional finite element Lagrange basis space.

Then, we will discuss how the computational domain is decomposed. The decomposition of the RDF method is similar to RBEM, using the generic approach. For the subdomains, archetypes $\hat{\Omega}_k, k = 1, \cdots, \hat{N}_\Omega$ are defined, and k^{th} archetype is instantiated N_m^k times. The instantiation of the k^{th} archetype is denoted as Ω_m^k, where m is the index of the subdomain. Thus, the total number of subdomains is $N_\Omega = \sum_{k=1}^{\hat{N}_\Omega} N_m^k$.

Moreover, since extra interface basis functions are added to the solution space, the interfaces are also treated as components of the system. Interface archetypes are defined as $\hat{\Gamma}^e, e = 1, \cdots, \hat{N}_\Gamma$. Then, $\hat{\Gamma}^e$ is transformed to an instantiation Γ_r^e, where r is the index of the instantiation. If we assume $\hat{\Gamma}^e$ is instantiated N_r^e times, the total number of interfaces is $N_\Gamma = \sum_{e=1}^{\hat{N}_\Gamma} N_r^e$.

Once the archetype components are defined, the global system can be assembled by instantiating the archetype subdomains and interfaces. To simplify the notation in our explanation, we condense the archetype and realization index into a single one. We assume $\Omega = \cup_{m=1}^{N_\Omega}\Omega_m$ and $\Gamma = \cup_{r=1}^{N_\Gamma}\Gamma_r$.

---

[10] As velocity fields are discussed here, we use $\mathbf{u}$ instead of u.

Then, basis functions defined on the archetype subdomains and interfaces are also instantiated correspondingly. That is,

$$\begin{aligned}\mathcal{V}_{m,0} &= \left\{ v_{m,i} = \underline{T}^k(\boldsymbol{\xi}_m^k) \circ v_i^k, v_i^k \in \mathcal{V}_0^k, k = 1, \cdots, \hat{N}_\Omega, m = 1, \cdots, N_m^k \right\}, \\ \Phi_r &= \left\{ \phi_{r,l} = \underline{T}^e(\boldsymbol{\xi}_r^e) \circ \phi_l^e, \phi_l^e \in \Phi^{\hat{\Gamma}^e}, e = 1, \cdots, \hat{N}_\Gamma, r = 1, \cdots, N_r^e \right\}.\end{aligned}$$

Therefore, the composed basis space for RDF systems is

$$\mathcal{U} := \bigcup_{m=1}^{N_\Omega} \mathcal{V}_{m,0} \oplus \bigcup_{r=1}^{N_\Gamma} \Phi_r, \tag{10}$$

where $\mathcal{V}_{m,0}$ are defined on $\overline{\Omega}_m$ with vanishing values on local boundaries, and Φ_r are defined on Γ_r.

The interface set of Ω_m is $\mathcal{I}_m$. Therefore, the local solution $u_m(\boldsymbol{\mu})$ can be represented by

$$u_m(\boldsymbol{\mu}) \approx \sum_{i=1}^{N_{\mathrm{RB},m}} a_{m,i}(\boldsymbol{\mu}) v_{m,i} + \sum_{\Gamma_r \in \mathcal{I}_m} \sum_{i=1}^{N_r} b_{r,i}(\boldsymbol{\mu}) \phi_{r,i}, \tag{11}$$

where $a_{m,i}$ and $b_{r,i}$ are the coefficients of the internal and interface RBs, respectively. $v_{m,i}$with $i = 1, \ldots, N_{\mathrm{RB},m}$ are basis vectors defined in Ω_m that vanish on local Dirichlet boundaries, and $\phi_{r,i}$ are basis vector instantiated on Γ_r that has in total N_r nodes. Thus, the system consists of two set of unknown coefficients, $a_{m,i}$ and $b_{r,i}$.

Then, the weak formulation of the problem is projected onto the cell-based and face-based basis functions, $v_{m,i}$ and $\phi_{r,i}$, respectively. We highlight that $v_{m,i}$ takes values in both Ω_m and $\partial\Omega_m$, while $\phi_{r,i}$ only takes values on the interface Γ_r.

For Ω_m, we can project Eq. 4 onto $\mathcal{V}_{m,0}$ and obtain

$$\begin{aligned}&(\dot{u}_m, v_{m,i}) + \underline{\mathcal{Q}}_\Omega(u_m, v_{m,i}) + \underline{\mathcal{Q}}_\Gamma(u_m, v_{m,i}) = \\ &\underline{\mathcal{F}}_m(v_{m,i}), \quad v_{m,i} \in \mathcal{V}_{m,0},\end{aligned} \tag{12}$$

where

$$\underline{\mathcal{Q}}_\Gamma(u_m, v_{m,i}) = \sum_{\Gamma_r \in \mathcal{I}_m} \sum_{j=1}^{N_r} b_{r,j} \underline{\mathcal{Q}}_\Gamma(\phi_{r,j}, v_{m,i}).$$

Note that, although $\phi_{r,i}$ is defined on Γ_r, it has values in the element adjacent to Γ_r[11]. Thus, even though $v_{m,i}$ vanishes on local boundaries, the interface contribution, $\underline{\mathcal{Q}}_\Gamma(u_m, v_{m,i})$, remains in the equation.

Then, we consider test space Φ_r at an interface $\Gamma_r = \Omega_m \cap \Omega_n$, Eq. 4 is reformulated as

$$\begin{aligned}&(\dot{u}_m, \phi_{r,l})|_{\Gamma_r} + (\dot{u}_n, \phi_{r,l})|_{\Gamma_r} + \underline{\mathcal{Q}}_\Gamma(u_m, \phi_{r,l}) + \underline{\mathcal{Q}}_\Gamma(u_n, \phi_{r,l}) = \\ &\underline{\mathcal{F}}_m(\phi_{r,l}) + \underline{\mathcal{F}}_n(\phi_{r,l}) \quad \phi_l \in \Phi_r.\end{aligned} \tag{13}$$

Be aware that Φ_Γ is only defined on Γ, so the volumetric terms are eliminated.

The algebraic form of RDF is formed by stacking Eqs. 12 and 13 for all subdomains and interfaces, as shown in Appendix A.2. The unknowns of the system are the coefficients of the internal and interface RBs, $a_{m,i}$ and $b_{r,i}$, respectively.

The RDF provides a general framework to achieve global approximations with local RBs. Iapichino et al. applied it for geometrical parameterization of an elliptic problem, the steady heat equation in the thermal fin [59, 195].

Martini et al. [196] derived a RDF-type reduced system while using a distinct strategy to construct reduced subspaces. The RBs are computed in two stages. The first step is to conduct a few global FOM simulations. Then, the interface modes are computed by extracting values at the internal interface $\phi_{r,l}(\boldsymbol{\mu}_j)$ of snapshots. Secondly, the interior RB is computed from a series of local problems with interface modes assigned as lifting functions at boundaries. Thus, the steady Eq. 12 defined in a subdomain is written by,

$$\underline{\mathcal{Q}}_\Omega(u, w_i) = \underline{\mathcal{F}}(w_i) - \underline{\mathcal{Q}}_\Gamma(\phi_{r,l}, w_i), \quad w_i \in \mathcal{W}_0 \text{ and } \phi_{r,l} \in \Phi.$$

Note that the last term in the right-hand side is formed from the standard lifting procedure and is thus known.[12] The internal and boundary RBs, as well as the velocity supremizer basis, are gathered and orthonormalized by the *Gram-Schmidt* process to produce the compact RBs. Finally, the coupled ROMs are solved in the frame of the RDF approach. The authors managed to model a steady-state problem (Stokes and Darcy equations) in a non-overlapping two-subdomain system, which considers geometric parameters.

4.2.3 Static Condensation Reduced Basis Element Method

Static Condensation Reduced Basis Element Method (SCRBEM) was proposed for domains assembled by non-overlapping archetype components. Similar to the RBEM, generic decomposition (Sect. 3.1.3) and localized training (Sect. 3.3.3) were employed in the methodology. As usual, the domain is decomposed $\Omega = \cup_{m=1}^{N_\Omega} \Omega_m$ with internal faces $\Gamma = \cup_{r=1}^{N_\Gamma} \Gamma_r$.

---

[11] This is a common practice in finite element methods, where basis functions defined on a node are extended to adjacent elements.

[12] This is equivalent to considering the second term in the left-hand side of Eq. 11 as known.

This approach is conceptually similar to the RDF method illustrated in the previous section. However, there are subtle differences, particularly in the definition and generation of the basis spaces, which will be explicitly highlighted later. The framework incorporates two key techniques: port reduction and static condensation. In the following sections, we will briefly discuss their fundamental principles and implementation.

4.2.3.1 Port Reduction RBEM The space is generated using a *port reduction* [197, 198] technique, which aims to separate the reduced basis into two parts: one for the surface and one for the inner zone. Those take the names of *port* and *bubble* space, respectively.

The port space Φ_r is defined only for interfaces Γ_r, and $\Phi_r \subset H^{\frac{1}{2}}(\Gamma_r)$. The bubble space is the range of basis vectors that vanish on all boundaries of the local domain $\mathcal{V}_{m,0} := \operatorname{span}\{v \in H^1(\Omega_m) \mid v|_{\partial\Omega_m} = 0\}$. Note that the definition of the basis $\mathcal{V}_{m,0}$ differs from Eq. 9, now the vectors are homogeneous on the whole $\partial\Omega_m$. With these changes, like in Eq. 10, the global test space is written as

$$\mathcal{W} := \bigcup_{m=1}^{N_\Omega} \mathcal{V}_{m,0} \oplus \bigcup_{r=1}^{N_\Gamma} \Phi_r. \tag{14}$$

We now present strategies to span Φ_r and $\mathcal{V}_{m,0}$. The reduced port and bubble subspaces can be generated from two different problems, as presented in [199].

The port training problem is sketched in Fig. 20a. It solves a two-subdomain problem[13] with parameterized Dirichlet BCs on $\hat{\Gamma}_{\text{in}}$ and $\hat{\Gamma}_{\text{out}}$. Finally, the solution at the interface $\hat{\Gamma}_{12}$ is extracted to form a snapshot matrix. This will be post-processed with a dimensional reduction technique to obtain the basis Φ_r.

The bubble RB is obtained by solving the single domain problem of Fig. 20b, in which the port basis functions are randomly weighted as boundary conditions $\hat{\Gamma}_{\text{in}}$, $\hat{\Gamma}_{\text{out}}$. Then, lifting functions[14] are utilized to homogenize Dirichlet BCs, and POD is utilized to compute the reduced bubble space. More descriptive training steps are presented in [199].

The trial space $\mathcal{U}$ is not equal to the test space $\mathcal{W}$. As the homogenization of the bubble space is achieved by lifting functions, the lifting functions should be involved to approximate the solution in Ω_m. Consequently, the trial space is defined as

$$\mathcal{U} := \bigcup_{m=1}^{N_\Omega} \mathcal{V}_{m,0} \oplus \bigcup_{r=1}^{N_\Gamma} \Phi_r \oplus \bigcup_{r=1}^{N_\Gamma} \Psi_r, \tag{15}$$

where $\Psi_r \subset H^1(\overline{\Omega}_m)$ are the lifting spaces used to homogenize the bubble space and defined for each interface Γ_r. Note that, as

Now, the local approximation is expressed by

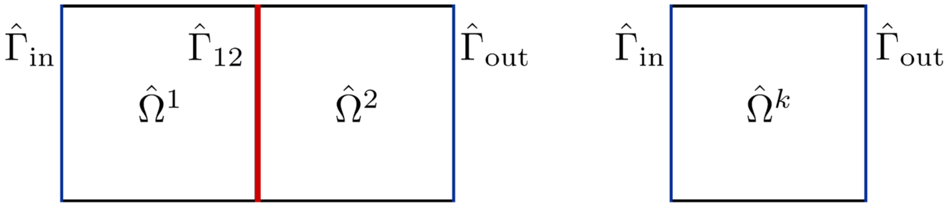
$$u_m(\boldsymbol{\mu}) \approx \sum_{i=1}^{N_{\text{RB},m}} a_{m,i}(\boldsymbol{\mu}) v_{m,i} + \sum_{\Gamma_r \in \mathcal{I}_m} \sum_{i=1}^{N_r} b_{r,i}(\boldsymbol{\mu}) \left(\phi_{r,i} + \psi_{r,i}\right),$$

where the coefficients $a_{m,i}$ and $b_{r,i}$ are the unknowns of the system, $v_{m,i}$ and $\phi_{r,i}$ are the basis vectors of $\mathcal{V}_{m,0}$ and Φ_r, respectively. $\psi_{r,i} \in \Psi_r$ is the lifting function. Since the lifting functions are computed with port modes as boundary conditions, the coefficients of the lifting functions are the same as the ones of the port modes.

By applying Galerkin projection of Eq. 4 to bubble $\mathcal{V}_{m,0}$ and port Φ_r spaces for Ω_m, we aim to find u_m that satisfy

$$\begin{aligned}
&(\dot{u}_m, v_i) + \underline{\mathcal{Q}}_\Omega(u_m, v_{m,i}) = \underline{\mathcal{F}}_m(v_{1,i}) && v_{m,i} \in \mathcal{V}_{m,0}, \\
&(\dot{u}_m, \phi_{r,l})\,|_{\Gamma_r} + (\dot{u}_n, \phi_{r,l})\,|_{\Gamma_r} + \underline{\mathcal{Q}}_\Gamma(u_m, \phi_{r,l}) && \phi_{r,l} \in \Phi_r, \\
&+ \underline{\mathcal{Q}}_\Gamma(u_n, \phi_{r,l}) = \underline{\mathcal{F}}_m(\phi_{r,l}) + \underline{\mathcal{F}}_n(\phi_{r,l}) &&
\end{aligned} \tag{16}$$

where, parameter $\boldsymbol{\mu}$ is omitted for brevity, and Ω_m and Ω_n are connected by Γ_r. This system, which looks exactly



(a) Port training problem.

(b) Bubble training problem.

Fig. 20 Port and bubble training. Redraw based on [199]. (**a**) The port training model for a generic $\hat{\Gamma}_{12}$ connects two reference partitions $\hat{\Omega}^1$ and $\hat{\Omega}^2$. Parametric Dirichlet BCs are assigned for Γ_{in} and Γ_{out}. The results of interface Γ_{mn} are collected as reduced port space. (**b**) The bubble training for $\hat{\Omega}^k$ with parameterized BCs $\hat{\Gamma}_{\text{in}}$ and $\hat{\Gamma}_{\text{out}}$. The BCs are random linear combinations of the port modes

---

[13] The authors [199] performed high-fidelity computations at the model.

[14] The authors [199] adopt solutions of Stokes equations as lifting functions for Navier-Stokes problems.

equal to Eqs. 12 and 13 represents a different reality, hidden in its test-trial spaces.

Benaceur et al. applied the aforementioned procedure to model steady-state Navier-Stokes flow (including heat transport) for 2D rivers, considering two physical parameters are considered, i.e., Reynolds number and Prandtl number [199]. Ebrahimi et al. [176] proposed a so-called *hyper-reduced RBEM*, which is similar to port-reduction RBEM. The novelty of their work is the implementation of the hyper reduction technique (*i.e., empirical quadrature procedure*) to further reduce the computational cost of the online stage. They applied the method to analyze a nonlinear heat transfer problem in a thermal fin.

4.2.3.2 Static Condensation We now illustrate the procedure of the so-called *static condensation*. The basic idea of this method is to divide a domain into inner and boundary components and then make the DoFs in the interior enslaved by the boundary DoFs. For example, let us consider the matrix system of Eqs. A41 for steady-state conditions. The internal DoFs can be eliminated considering $\mathbf{a} = \underline{\underline{C}}^{-1}(\underline{\underline{F}}_{\Omega} - \underline{\underline{D}}\mathbf{b})$, to get the condensed system,

$$(\underline{\underline{G}} - \underline{\underline{Q}}\,\underline{\underline{C}}^{-1}\underline{\underline{D}})\mathbf{b} = \underline{\underline{F}}_{\Gamma} - \underline{\underline{Q}}\,\underline{\underline{C}}^{-1}\underline{\underline{F}}_{\Omega},$$

where only DoFs $\mathbf{b}$ in the boundary remain, and the internal DoFs are expressed in terms of the boundary ones. Note that for time-dependent problems, e.g., Eq. 16, the implementation of static condensation is similar to the steady-state case after discretization in temporal domain.

Patera et al. [200, 200] integrated the static condensation technique into the port reduction RBEM, which leads to the so-called SCRBEM. Since the internal DoFs are enslaved by the boundary ones, the online stage only needs to solve for the boundary DoFs, which significantly reduces the computational cost. We have illustrated the implementation procedure of the technique for a steady-state case in Appendix A.4, including computation of RBs and the constructing of the condensed reduced system.

Patera et al. have applied the static condensation RBEM to a variety of problems. For instance, they applied the framework to simulate large-scale acoustic devices governed by the Helmholtz equation [200]. Henceforth, a posteriori error estimation was studied in [198]. Their research also includes numerical results for the heat transfer [201] and a linear elasticity problem [47]. In collaboration with Vallaghee [202], the technique is utilized for analysis of a conjugate heat exchanger model containing one dimensional (1D) bulk fluid and solid heat conduction.

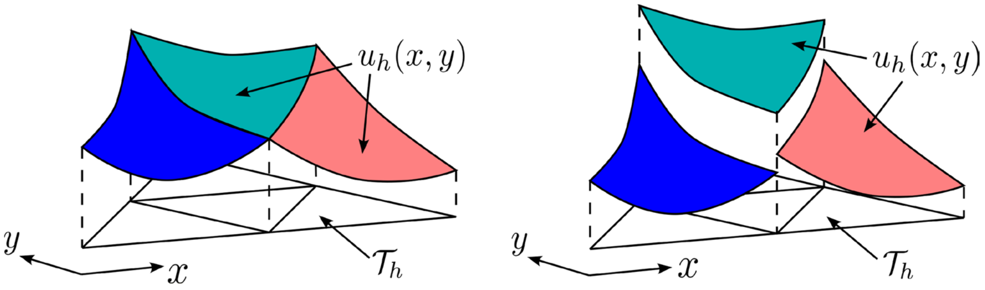
We also found contributions from other researchers. For example, Kerfriden et al. used it for the analyses of nonlinear fracture mechanics [203]. A more recent publication by Mu et al. employed it for linear steady-state convection-diffusion equations with random diffusivity and velocity [46].

4.2.4 Discontinuous Galerkin-Based Local ROM Methods

4.2.4.1 Basic Formulation of Discontinuous Galerkin The *Discontinuous Galerkin* (DG) method was originally proposed as a numerical technique for enforcing the Dirichlet condition in elliptic equations (i.e., *Nitsche method* [204]) and for the numerical solution of neutron transport [205, 206]. Then, Cockburn, Shu, Riviere, and et. al. [207–210] have further developed it into a general framework for solving PDEs, which combines features of the FE and f inite volume methods. Demkowicz et al. [211–215] have contributed to the development of the *discontinuous Petrov-Galerkin* method, which uses optimal test functions[15] to achieve stability and accuracy. Note that DG is becoming increasingly popular as a high-order method for solving PDEs. Interested readers can refer to the aforementioned references, the books [216–218], and the references therein for more details.

The standard Galerkin method (also known as continuous Galerkin) used in FE employs continuous basis functions across element interfaces. Unlike it, DG utilizes discontinuous basis functions, as illustrated in Fig. 21. To minimize variable jumps across interfaces, continuity and

Fig. 21 Schematic description of the basis functions employed for continuous Galerkin (left) and discontinuous Galerkin (right). The high fidelity element $\mathcal{T}_h$ and basis function $v_h\{\mathbf{x}, \mathbf{y}\}$. Redrawn based on [219]



---

[15] As a Petrov-Galerkin method, the trial functions are not equal to the test functions, and the test space is designed to optimize the stability and accuracy of the numerical solution.

smoothness constraints are incorporated into the governing equations as penalization terms. The high-fidelity formulation also includes numerical flux terms defined on the interfaces. These also remain in the projection-based ROMs. Therefore, non-overlapping partitions can be *glued* together directly in the online stage without Lagrange multipliers.

The formulation depends on the exact equation to be approximated. Here, we use a Laplacian equation. Its weak formulation reads

$$-\int_{\Omega} \nabla u \cdot \nabla w + \int_{\partial\Omega} \nabla u w \cdot \mathbf{n} = 0,$$

where u is the solution and w is the test function. To ensure sufficient regularity for the DG formulation such that traces on the boundaries are well-defined, we assume $u \in H_0^1(\Omega) \cap H^2(\Omega)$. Setting the test functions to be discontinuous across element interfaces, we take $w \in H^1(\mathcal{T}_h) = \{w \in L^2(\Omega) : w|_K \in H^1(K)\ \forall K \in \mathcal{T}_h\}$, which is a broken Sobolev space defined on the mesh partition. To derive the DG formulation, we define on an interface Γ_r, which has two sides, denoted by $-$ and $+$, and the jump operator[16],

$$[w]|_{\Gamma_r} = w^- - w^+,$$

and average operator,

$$\{w\}|_{\Gamma_r} = \frac{1}{2}\left(w^- + w^+\right),$$

where the two subscripts $-$ and $+$ denote the values at the two sides of Γ_r, indeed, from the two adjacent subdomains of Γ_r.

As usual, we assume the entire domain is decomposed into N_Ω subdomains $\Omega = \cup_{m=1}^{N_\Omega} \Omega_m$, and the interface is collected into a set Γ. The integral over the whole model results in

$$-\sum_{\Omega_m \in \Omega} \int_{\Omega_m} \nabla u \cdot \nabla w + \sum_{\Gamma_r \in \Gamma} \int_{\Gamma_r} [\nabla u w] = \underline{\mathcal{F}}(w),$$

where $\nabla u w$ is a vector, and the jump operator is defined as $[\nabla u w] = \nabla u^- w^- \cdot \mathbf{n}^- + \nabla u^+ w^+ \cdot \mathbf{n}^+$. The global boundary conditions are involved in the right-hand side, i.e., $\underline{\mathcal{F}}(w)$.

Remark that the term defined on the interface, e.g., $\sum_{\Gamma_r \in \Gamma} \int_{\Gamma_r} [\nabla u w]$, is called *numerical flux* in the DG framework. Various schemes for the numerical flux have been discussed, and interested readers can turn to the references [209, 220, 221] for more solid explanations. Here, we illustrate a typical strategy for dealing with it.

By using the relationship $[uw] = [u]\{w\} + \{u\}[w]$, we have

$$-\sum_{\Omega_m \in \Omega} \int_{\Omega_m} \nabla u \cdot \nabla w + \sum_{\Gamma_r \in \Gamma} \int_{\Gamma_r} \left([\nabla u]\{w\} + \{\nabla u\}[w]\right) = \underline{\mathcal{F}}(w).$$

Because we assumed the exact solution has higher regularity $u \in H_0^1(\Omega) \cap H^2(\Omega)$, both u and its exact normal flux, $[\nabla u] = \nabla u^- \cdot \mathbf{n}^- + \nabla u^+ \cdot \mathbf{n}^+ = 0$, are continuous across the interfaces. Consequently, the jump terms evaluate to zero for the exact solution, i.e.,

$$[u] = 0 \text{ and } [\nabla u] = 0,$$

which implies that the additional contribution $\sum_{\Gamma_r \in \Gamma} \int_{\Gamma_r} [\nabla u]\{w\}$ sums to zero. Thus, we get

$$\begin{aligned} &-\sum_{\Omega_m \in \Omega} \int_{\Omega_m} \nabla u \cdot \nabla w \\ &+ \sum_{\Gamma_r \in \Gamma} \int_{\Gamma_r} \{\nabla u\}[w] = \underline{\mathcal{F}}(w). \end{aligned} \tag{17}$$

Two terms are added to the left hand side of Eq. 17 to improve its numerical features: (i) $\frac{\gamma}{h} \int_\Gamma [u][w]$ to penalize the jump of u, with γ a factor and $h = \max\{|\mathbf{x} - \mathbf{y}|, \forall \mathbf{x}, \mathbf{y} \in \tau\}$ the largest diameter of the element in the mesh; (ii) A symmetric term $\varepsilon \int_\Gamma \{\nabla w\}[u]$ where the constant $\varepsilon \in \{-1, 0, 1\}$, to create a formulation which is symmetric or not depending on the value of ε. The total results in

$$\begin{aligned} &-\sum_{\Omega_m \in \Omega} \int_{\Omega_m} \nabla u \cdot \nabla w + \\ &\sum_{\Gamma_r \in \Gamma} \int_{\Gamma_r} \left(\{\nabla u\}[w] + \varepsilon \{\nabla w\}[u] + \frac{\gamma}{h} \int_{\Gamma_r} [u][w] \right) = \underline{\mathcal{F}}(w). \end{aligned} \tag{18}$$

The above derivation can be extended to more complex problems, such as parabolic, Stokes, and Navier-Stokes equations. The major concern of the implementation is dealing with different terms, e.g., convection terms, and reformulating them as numerical flux regarding the jump and average conditions on interfaces. The detailed procedures and discussions are presented in [209, 221].

Moreover, we want to highlight that the DG method originates from the Nitsche method. The three terms $\frac{\gamma}{h} \int_\Gamma [u][w]$,

---

[16] For a vector $\mathbf{w}$, the jump operator is defined as $[\mathbf{w}] = \mathbf{w}^- \cdot \mathbf{n}^- + \mathbf{w}^+ \cdot \mathbf{n}^+$, where $\mathbf{n}^-$ and $\mathbf{n}^+$ are the outward normal vectors of the two adjacent subdomains of Γ_r.

$\int_\Gamma \{\nabla u\}[w]$, and $\varepsilon \int_\Gamma \{\nabla w\}[u]$, penalty, consistency, and symmetry terms, respectively, were initially proposed by Nitsche [204] to weakly enforce the Dirichlet condition when solving the Poisson problem with FEM.

Thus, we may consider that the DG discretization of the PDEs is constructed by applying the Nitsche formulation to each element, and the continuity of the solution across the element interfaces is enforced by adding penalty terms to the weak formulation. The consistency and symmetry terms are also added to ensure the accuracy and stability of the method. See [222] for a comprehensive description about utilizing Nitsche method for variation problems.

Due to the characteristics of the Nitsche formulation for treating boundary conditions, it has been widely used to solve problems considering interfaces, e.g., elliptic interface problem [223–225], fluid-structure interaction [226–229], and isogeometric analysis [230–232]. For brevity, we won't go into more detail here. Interested readers can refer to the above references for more information.

4.2.4.2 Discontinuous Galerkin Reduced Basis Element Method The DG approach integrating local RBs was proposed by Antonietti et al. [23] as a generalization and an improvement of both RBHM and RDF, called *Discontinuous Galerkin Reduced Basis Element Method* (DGRBEM). It follows the philosophy of RBEM in domain decomposition, adopting non-overlapping subdomains and a generic decomposition strategy (See Section 3.1.3). The offline high-resolution DG simulations are performed in a generic partition, which is referred to as a local training strategy (discussed in Sect. 3.3.3).

As RB functions are computed for each archetype, they are not matched on the interfaces satisfying the basic configuration of DG. Indeed, the local RBs, containing both internal and surface features, e.g., $\mathcal{V} \in H^1(\Omega)$, can be directly adopted for Eq. 18. Hence, every local solution can be approximated as a linear combination of local modes. We believe the derivation of a DGRBEM system and its matrix form is not complicated. Therefore, we don't clarify the details in this review.

Antonietti et al. verified their algorithm with elliptic problems for multiple-division models, in which both physical properties and geometrical shapes are parameterized [23]. Pacciarini et al. [233] extended the DGRBEM for parametrized Stokes problems, in which high-fidelity snapshots were collected by utilizing spectral element methods. A recent research from Chung et al. [234] demonstrates the same methodology on three linear PDEs: the Poisson equation, the Stokes flow equation, and the advection-diffusion equation.

We highly suggest that readers refer to an up-to-date investigation regarding *Friedrichs' systems*[17] released by Romor et al. [235]. The general technique is tested and validated by a series of PDEs, i.e., Maxwell equations, compressible linear elasticity, advection diffusion reaction, and incompressible Navier-Stokes equations. Note that the authors involved graph neural networks to efficiently infer the vanishing viscosity solution that is challenging for projection-based approaches.

Since the snapshots are collected from high-fidelity simulations at the subdomain level, the boundary conditions of local problems should be parameterized to approximate the global problems. Different types of boundaries can be imposed: non-homogeneous Neumann is adopted by Antonietti et al. [23] and Pacciarini et al. [233], while non-homogeneous Dirichlet is employed by Chung et al. [234].

4.2.4.3 Local POD Discontinuous Galerkin ROM Note that the strategy for computing RBs in frames of DG is not unique. Ferrero et al. [236] developed a *local POD-DG* algorithm, whose formulations are the same as the DGRBEM except for domain decomposition and local RBs generation [237, 238]. The approach considers individual decomposition of the whole problem and builds local POD RBs for each partition utilizing global solutions (as addressed in Sect. 3.3.2). Indeed, it can be regarded as a simplification of DGRBEM. As illustrated by Ferrero [236], the method results in good accuracy and significant acceleration for the 2D Euler and Reynolds averaged Navier-Stokes equations with the Spalart-Allmaras turbulence closure.

We highlight that the DG-based systems for FOMs and ROMs are consistent except for basis functions. This entitles DG to support the coupling of hybrid systems involving FOMs and ROMs, as presented by Riffaud et al. [24, 239]. They applied the method to model the isentropic vortex governed by 2D unsteady Euler equations, in which the computational domain is decomposed into non-overlapping individual subdomains. They also managed to couple ROMs and FOMs for unsteady transonic flows in a converging-diverging nozzle and over a NACA0012 airfoil in the presence of shocks [24, 239]. They use a high-fidelity model to represent regions of complex nonlinear physical phenomena and a ROM to approximate elsewhere efficiently, with which the computational complexity is significantly reduced and the accuracy is comparable to high-resolution solvers.

A series of articles by Ohlberger, Kaulmann et al. proposed and investigated their *localized reduced basis multiscale* method. Although the algorithm is implemented

---

[17] The Friedrichs' systems are regarded as a unified framework for various PDEs [235, 365]: first and second order uniformly hyperbolic, elliptic and parabolic PDEs.

for different equations and applications, the methodology and procedures are almost the same as Ferrero's method. The high-resolution DG is utilized to generate snapshots, and individual RBs are computed for each partition. It was originally introduced for general parametrized elliptic problems [240]. Then, it is improved to solve a two-phase flow in porous media [241, 242] and scalar parabolic evolution equations [45]. The posteriori *error estimations* are detailed for elliptic problems in [243] and parabolic equations in [45]. The online enrichment technique[18] is also included in their framework, and the details are discussed in [243, 244].

4.2.5 Partition of Unity Method and ROM

The *Generalized Finite Element Method* (GFEM) is an extension of the classic FEM [245], which adopts a standard functional basis[19] to approximate the common region of a problem and an enriched basis to capture specific local phenomena. The coupling is achieved following the *Partition of Unity Method* (PUM) [246, 247], which glues local bases defined in different regions to construct a global basis.

The methodologies of GFEM and PUM can be incorporated within the framework of the local ROM. They are utilized for problems consisting of overlapping divisions. The local RBs are computed using either local or global solutions. Obviously, they should not be continuous along interfaces. PUM is employed to weigh the local RBs and produce a continuous global space. Once the global RB is available, the reduced system 5 can be easily constructed and solved.

Let us suppose we have a series of local RBs, spanned by the components $\{v_{m,i}\}_{m=1}^{N_\Omega}$. The i^{th} vector of the PUM space can be defined as

$$v_i^{\text{PUM}} := \text{span}\left\{\bigcup_{m=1}^{N_\Omega} \underline{\psi}_m v_{m,i}\right\},$$

where $\underline{\psi}_m$ is a PoU function defined for Ω_m. Consequently, we can form a global reduced space

$$\mathcal{V}^{\text{PUM}} := \left\{\sum_{i=1}^{N_{\text{RB}}} v_i^{\text{PUM}}\right\} \subset H_0^1(\Omega).$$

To finalize this method, we just need to define the $\underline{\psi}_m$. We highlight that the formulation of the *Partition of Unity* (PoU) functions is not unique. Various examples are illustrated in [246, 247], as well as in the book [248]. We will now give a basic description of the definition and construction of the PoU functions. Firstly, they satisfy

$$0 \le \underline{\psi}_m\Big|_{\Omega_m} \le 1,\quad \underline{\psi}_m\Big|_{\Omega\setminus\Omega_m} = 0,\quad \sum_{m=1}^{N_\Omega} \underline{\psi}_m(\mathbf{x}) = 1,\quad \text{and}\quad \left\|\nabla\underline{\psi}_m\right\|_{L^\infty(\Omega)} \le C_m,$$

where C_m is a constant. Moreover, readers can refer to Appendix B.2 for an example for the construction of the PoU functions for 1D problems.

The methodology of GFEM and its applications for constructing ROMs are presented in the references below. Babuška et al. [249] proposed the method to solve second-order elliptic PDEs. They also discussed the procedure to identify optimal local approximation spaces. Additionally, they integrated into the method the *oversampling* strategy to generate local solutions [50, 148] and tested the approach for elliptic PDEs with random boundary conditions. Baiges et al. adopted the strategy to couple bases for the construction of local ROMs and/or FOM-ROM systems to model incompressible Navier-Stokes flow around cylinders [141].[20] They compared the performance of two spaces, i.e., localized global RBs (Sect. 3.3.1) and subdomain-level RBs (Sect. 3.3.2). Schleuß et al. extended the procedure to analyze parabolic problems, proposing local space-time approximation spaces [250].

A recent contribution from Smetana et al. [146] employed it to deal with nonlinear elliptic problems, considering *generic decomposition* (Section 3.1.3) with overlaps and *oversampling training* (Sect. 3.3.3). The work also involved an adaptive basis enrichment algorithm when seeking global approximations in the online phase. They adopted an alternative formulation for the construction of the PoU functions

$$\underline{\psi}_{m,i}(\mathbf{x}) = \frac{v_{m,i}(\mathbf{x})}{\sum_{n=1}^{N_\Omega} v_{n,i}(\mathbf{x})},\qquad \mathbf{x}\in\Omega, m = 1,\cdots,N_\Omega,$$

where $v_{m,i}$ is a RB vector defined in Ω_m and $\mathbf{x}$ denotes the global coordinate. Be aware that the subdomains are overlapping, thus $\sum_{n=1}^{N_\Omega} v_{n,i}(\mathbf{x}) \neq v_{m,i}(\mathbf{x})$ in the shared regions.

4.2.5.1 Multiscale Finite Element Method A technique similar to the ideology of the previous approach is the so-called *Multiscale Finite Element Method* (MsFEM). The MsFEM employs a local basis to represent especially small-

---

[18] This technique dynamically updates the reduced basis during the online phase. This improves capturing new parametric solutions that are deficiently represented by the current RB generated in the offline stage. Instead of relying solely on a precomputed reduced space, the ROM can identify regions where additional basis functions are needed and incorporate them adaptively. This helps achieve accuracy whilst remaining efficient.

[19] It is typically a polynomial basis.

[20] The authors combined several local bases to build a global basis, and the methodology is similar to PUM, although they didn't explicitly mention the PUM in the text.

scale features inside of a large-scale solution [145]. As clarified in [251], MsFEM can be divided into two steps: (i) construct a subdomain-level basis that embeds essential multiscale information and (ii) couple the local bases to form a global formulation.

From this description, one may conclude that MsFEM and GFEM have similarities. Thus, it often happens that MsFEM and GFEM are discussed together in the literature [50, 148, 249]. Resemblance appears in the following aspects: (i) the two are based on FEM and can be regarded as an extension of the method to enhance computational efficiency; (ii) the local basis is capable of capturing specific variations in subregions; (iii) the basis functions are continuous across the interfaces. Thus, one can resort to the general weak formulation to solve the system.

In the early stage of MsFEM, researchers using this approach did not regard this technique as a ROM method [145]. However, it was later recognized as a form of local ROM [252, 253]. We believe that the reason is that: (i) the entire domain is decomposed into subdomains; (ii) the local RBs are generated by solving local problems, which is a common strategy for computing RBs; (iii) the global solution is approximated by a linear combination of the local RBs. It is obvious that the MsFEM follows the general procedure of a local ROM method. Therefore, it is reasonable to consider it as a local ROM method. The following paragraph will give a brief introduction to the strategy for constructing the continuous basis.

Here, we adopt a linear elliptic equation for explanation and refer the readers for other examples, insight, and further discussion to [251]. The problem adopted is

$$\nabla \cdot (k(\mathbf{x})\nabla u) = f, \quad \text{in } \Omega,$$

where $k(\mathbf{x})$ is the conductivity varying over multiple scales.

The domain is now spatially discretized into two scales, high-fidelity elements e and subdomains Ω_m.[21] The aim is to compute basis functions in Ω_m and represent global solutions via the RB of Ω_m instead of the basis of e. The standard FEM basis vectors are denoted by ϕ_i^e, e.g., Lagrange polynomials. As usual, $\phi_{m,i}, i = 1, \ldots, N_m^{\mathrm{RB}}$ denote basis functions in Ω_m. We compute each $\phi_{m,i}$ as the solutions of

$$\nabla \cdot (k(\mathbf{x})\nabla \phi_{m,i}) = 0, \quad \text{in } \Omega_m,$$

subjected to BCs

$$\phi_{m,i} = \phi_i^e, \quad \text{on } \partial\Omega_m,$$

where ϕ_i^e is the i^{th} standard FEM basis function defined on the fine elements. The BCs are crucial for the construction of the RB, as they ensure the continuity of the basis functions across interfaces when assembling the entire domain. Note that the basis spanned by $\phi_{m,i}$ is also referred to as a *multi-scale* basis [251].

The solution in each block is approximated with N_m^{RB} vectors as $u_m = \sum_{i=1}^{N_m^{\mathrm{RB}}} a_{m,i}\phi_{m,i}$. Also, $\mathcal{V}_m \equiv \mathcal{W}_m \equiv \mathrm{span}\{\phi_{m,i}\}$. We recall that the BCs of $\phi_{m,i}$ are ϕ_i^e. Since the ϕ_i^e are continuous as required by standard FEM, $\phi_{m,i}$ are conforming on the interfaces when assembling the entire domain. More precisely, the global $\phi_i = \cup_m \phi_{i,m}$ are continuous in every interface. Indeed, $\mathcal{V} \in H_0^1(\Omega)$

The MsFEM is to find $u \in \mathcal{V}_m$ such that

$$\sum_m \int_{\Omega_m} k(\mathbf{x})\nabla u \cdot \nabla \phi_{m,i} = \int_{\Omega} f\phi_{m,i}, \quad \forall \phi_i \in \mathcal{W}_m.$$

The MsFEM was developed by Hou et al., along with contributions from Yalchin Efendiev. It was applied to several problems, e.g., to linear and nonlinear elliptic problems [251, 254, 255]. There are alternative procedures to construct the RB. The oversampling technique (see Sect. 3.3.3) was initially utilized for solving local problems. Then, the framework was extended to develop the *generalized MsFEM* for elliptic PDEs [252, 253, 256]. This reduces the cost of constructing the RB when considering a complex input space. Henning et al. proposed an alternative method to generate the multiscale space $\mathcal{V}_m$, namely the localized orthogonal decomposition [257, 258]. Their approach was applied to elliptic PDEs with inhomogeneous Dirichlet and Neumann boundary conditions. The RB functions can be constructed adaptivity using error indicators to obtain larger accuracy for nonlinear problems [259].

4.2.6 Optimization-Based Algorithms

We now address the method based on a functional optimization strategy. Contrary to Sect. 4.2.1, here we do not utilize the Lagrange multipliers procedure to deduce additional equations to add to the system. In the methods discussed below, an optimization problem is properly resolved.

4.2.6.1 Optimization-Based Standard Galerkin The optimization-based domain decomposition technique is accomplished by equalizing variables on both sides of the interfaces. At the same time, the governing equations and

---

[21] The elements and blocks are not overlapping. They are called fine/micro elements e and coarse/macro blocks Ω_m in the frame of MsFEM.

boundary conditions are satisfied. This methodology is intended for non-overlapping domains.

We follow the standard procedure to obtain $\Omega = \cup_{m=1}^{N_\Omega} \Omega_m$, $\Gamma = \cup_{r=1}^{N_\Gamma} \Gamma_r$, and the interfaces between Ω_m and Ω_n is noted as $\Gamma_{[mn]} = \{\Gamma_r | \Gamma_r = \partial\Omega_m \cap \partial\Omega_n, \Gamma_r \subset \partial\Omega_m\}$. For this methodology, we do not establish special restrictions on the local basis of the trial and test spaces. We refer the readers to any of the procedures described above for the derivations of the RBs spanning those spaces.

We define a general target function $\underline{\mathcal{J}}$ on $\Gamma_{[mn]}$, which is minimized to obtain a global solution,

$$\underline{\mathcal{J}}(u_m, u_n; g) := \frac{1}{2}\int_{\Gamma_{[mn]}} |u_m - u_n|^2 + \frac{\gamma}{2}\int_{\Gamma_{[mn]}} |g|^2 . \qquad (19)$$

Here, the first term denotes a jump of u across the interface. The second term is known as the *regularization* term. It arises as a way to impose the homogeneity of a secondary property in the interface and to ensure *well-posedness*. Clearly, the parameter $\gamma > 0$ controls the relative importance of the terms of Eq. 19. We will give more details regarding g below.

The optimization-based approach aims to find a solution that minimizes the target function. It should also satisfy the local governing PDEs Eq. 4 and pertinent BCs. Consequently, the problem is reformulated as finding u such that

$$\min \underline{\mathcal{J}}(u_m, u_n; g) \quad \text{s.t.} \begin{cases} \sum_{m=1}^{N_\Omega} \underline{\mathcal{R}}_m(u, w) = 0, \quad \forall (u_m, w_m) \in (\mathcal{V}, \mathcal{W}), \\ u = g_D \text{ in } \Gamma_D, \text{ and } \partial u / \partial \mathbf{n} = g_N \text{ in } \Gamma_N, \end{cases} \qquad (20)$$

where $\begin{array}{l}\underline{\mathcal{R}}_m(u_m, w_m) := (\dot{u}_m, w_m) + \underline{\mathcal{Q}}(u_m, w_m) \\ -\underline{\mathcal{F}}(w_m), \quad (u_m, w_m) \in (\mathcal{V}_m, \mathcal{W}_m)\end{array}$ is the residual of the governing equations in Ω_m. The first term of $\underline{\mathcal{R}}_m$ is only for time-dependent problems, and $\underline{\mathcal{Q}}(u_m, w_m)$ denotes the spatial operator. The second line of Eq. 20 indicates that the solution should satisfy the global BCs.

Prusak et al. applied the methods for two incompressible Navier-Stokes benchmarks: the stationary backward-facing step and lid-driven cavity flow [48, 58]. The authors utilized Eq. 19 with a regularization term $\mathbf{g}$ that accounts for viscous stress tensor on the surface. Follow the ideas of Gunzburger [260], the term is defined as

$$\underline{\mathcal{J}}(\mathbf{u}_m, \mathbf{u}_n; \mathbf{g}) := \frac{1}{2}\int_{\Gamma_{[mn]}} |\mathbf{u}_m - \mathbf{u}_n|^2 + \frac{\gamma}{2}\int_{\Gamma_{[mn]}} |\mathbf{g}|^2 , \qquad (21)$$

where $\mathbf{u}$ is the velocity field, $\mathbf{g} = \left(\nu \frac{\partial \mathbf{u}}{\partial \mathbf{n}} - p\mathbf{n}\right)\Big|_{\Gamma_{[mn]}}$ is the flux across the interface and meanwhile the regularization term, ν is the viscosity, $\mathbf{n}_{\Gamma_{[mn]}}$ is the normal vector of $\Gamma_{[mn]}$, p_m is the pressure, and γ is a user-defined regularization parameter.

In this study, global solutions were collected and then divided into non-overlapping subdomains to generate local RBs (see Sect. 3.3.2). Both FOM and ROM problems were solved by a *gradient-based optimization algorithm*. The exact optimality system is reformulated in terms of a Lagrangian functional, which is detailed in the aforementioned references, as well as in [260].

Taddei et al. followed a similar strategy to analyze flow dynamics (Navier-Stokes equations) in blood vessel shape systems that are assembled by two archetype components [261]. They suggested penalizing the discontinuity of both velocity $\mathbf{u}$ and pressure p fields. They utilized a modified regularization term. This results in the following optimization function

$$\begin{aligned}\underline{\mathcal{J}}\left(\mathbf{u}_m, \mathbf{u}_n, p_m, p_n; \mathbf{g}, h, \nabla_{\Gamma_{[mn]}}\mathbf{g}\right) := &\frac{1}{2}\int_{\Gamma_{[mn]}} |\mathbf{u}_m - \mathbf{u}_n|^2 + \frac{1}{2}\int_{\Gamma_{[mn]}} |p_m - p_n|^2 \\ &+ \frac{\gamma}{2}\int_{\Gamma_{[mn]}} \left(|\nabla_{\Gamma_{[mn]}}\mathbf{g}|^2 + |\mathbf{g}|^2 + |h|^2\right),\end{aligned}$$

where $\mathbf{g}$ is given in Eq. 21, $h|_{\Gamma_{[mn]}} = \frac{\partial p}{\partial \mathbf{n}}\Big|_{\Gamma_{[mn]}}$, and $\nabla_{\Gamma_{[mn]}}\mathbf{g}$ denotes the gradient of $\mathbf{g}$ in the tangential direction of $\Gamma_{[mn]}$.

To generate RBs for interface control, they proposed a *pairwise-training* approach. They performed high-fidelity simulations for systems containing two partitions. Since each archetype might have several inlets and outlets, they construct various small systems to represent all possible connections between the two archetypes. The two-subdomain systems were resolved considering random Dirichlet and Neumann BCs. Also, they were parameterized with a range of Reynolds numbers and shapes. The internal RBs are constructed considering localized training (See Sect. 3.3.3). Adaptive enrichment procedure[22] was also integrated into the framework.

Finally, we summarize here a method that can be classified to this typology, but that is a combination of several of the methodologies described in this document. Sambataro et al. developed an innovative scheme named *one-shot overlapping Schwarz* method [25, 142]. They reformulated the standard iterative Schwarz procedure (see Sect. 4.3) to be an optimization strategy that minimizes the lump sum of interface jumps.[23] The subdomains are represented through *port*

---

[22] This approach continuously evaluates the error of the ROM and enriches the reduced basis when necessary. New FOM simulations are carried out for the worst predicted parametric data points, and new modes are added to improve approximation accuracy.

[23] Although the authors use the name *Schwarz* that is regarded as an iterative approach, we categorize their method into the group after analyzing their methodology.

and *bubble* spaces (see Sect. 4.2.3). Those are defined to represent interiors and internal boundaries. Then, the *static condensation* procedure is utilized to eliminate the interior DoF (see Section 4.2.3). Therefore, the global approximation is reconstructed by a reduced system involving only interface DoFs.

4.2.6.2 Least Squares Petrov-Galerkin The second optimization-based procedure we discuss is based on the *least squares Galerkin* method [12, 53, 262]. In this method, the residual of the governing equations is minimized in a least squares sense. Consequently, the ROM for a domain decomposition problem can be formulated with extra constraints at the interfaces.

It must be emphasized, in the standard optimization-based Galerkin, the optimization function enforce the interface continuity, while the governing equations are constraints. However, in the least squares Galerkin method, the equation residual is minimized, and the interface equality is a constraint. Therefore, the optimality system is written as

$$\min \frac{1}{2} \sum_{m=1}^{N_\Omega} \left\| \underline{\mathcal{R}}_m \left(u_m^\Omega \cup u_m^\Gamma, w_m \right) \right\|_2^2, \tag{22}$$

subject to the restrictions,

$$\sum_{\Gamma \in \partial\Omega_m \cap \partial\Omega_n} \int_\Gamma \left| u_m^\Gamma - u_n^\Gamma \right| = 0,$$

where u_m^Ω and u_m^Γ denote interior and local surface values, respectively. $\underline{\mathcal{R}}_m$ gives the residual of the governing equations in Ω_m (see Eq. 20). The constraints are utilized to impose equality at interfaces (or overlapping regions if they exist) of two neighbouring subdomains Ω_m and Ω_n.

Hoang et al. have utilized the technique for parameterized Laplacian and Burgers' equations in a non-overlapping partitioned computational domain [175]. This study has also incorporated several topics thoroughly. Notably, it discusses the utilization of strong or weak equality constraints on the interfaces. The performance of different strategies to build RBs was also investigated. Specifically, the behavior of RBs constructed with separated interior/boundary or full-subdomains, see Sect. 3.3.4. The benefits of hyper-reduction are also revealed in their analysis. The authors adopted two training procedures to generate snapshots, one that requires full-system solutions (Sect. 3.3.2) and a second that only demands data from a single subdomain (see Sect. 3.3.1).

Note that this methodology is prone to be integrated into a framework of Neural Networks to solve the reduced system, as investigated in [54, 172].

4.2.7 Multiphysics Problems

Multiphysics models are typically governed by two or more PDEs, which might necessitate partitioning to account for several subregions. ROMs are extensively applied to simulate these phenomena. However, the coupling of several physics is different from a single continuum. Considering the distinct characteristics, a separate description of the topic is presented below.

4.2.7.1 Fluid-Structure Interaction Among multiphysics, *FSI* is especially relevant in many practical applications. There are two widely used approaches for solving the high fidelity FSI: fully-coupled/monolithic methods and partitioned/iterative methods [263]. More details can be seen from [9]. We will present the former here, and the latter will be described in Sect. 4.3.

According to the observed references, *Arbitrary Lagrangian Eulerian* (ALE) method is the most frequently applied for solving high-fidelity FSI problems [264]. The formulation incorporates the fluid mesh displacement as an additional variable, enabling the fluid domain to follow the deformation of structures. Subsequently, solutions from the two regions are collected as snapshots, including displacement and velocity for both fields, as well as fluid pressure. Readers can refer to [9, 264] for more detailed descriptions regarding ROMs for FSI problems.

Before applying Galerkin projection to construct the intrusive ROM, a RB should be computed. Although the FSI problems consist of two physics, one can compute a global reduced basis for the entire model. For example, the displacement fields[24] can be collected into a single snapshot matrix,

$$\underline{\underline{S}}^d = \left[[\mathbf{d}_1^f, \mathbf{d}_1^s]^{\mathrm{T}}, \ldots, [\mathbf{d}_{N_\mu}^f, \mathbf{d}_{N_\mu}^s]^{\mathrm{T}} \right],$$

where $\mathbf{d}^f$ and $\mathbf{d}^s$ are the fluid and solid displacements, respectively. N_μ is the number of snapshots. The RB can be computed by applying POD to $\underline{\underline{S}}^d$, and the resulting basis functions contain information from both fluid and solid.

The same strategy can be used for other variables. Thus, due to the continuity of high-fidelity solutions, the resulting basis functions ensure continuity across the interface. The RBs can construct a monolithic reduced system. Liberge et al. applied the method for a transient flow around an oscillating cylinder [265]. Ballarin and Rozza extended it for more general parameterization cases for a coupling of Navier-Stokes flow and linear elastic structures [266].

---

[24] The displacements of the mesh corresponds to the spatial coordinates, so they should be vectors.

Note that the above investigations employing a global RB for two systems are not strictly regarded as local ROM approaches. We begin discussing studies that consider two separate local RBs for fluid and solid, respectively. An example is presented to clarify the coupling.

Given two sets of governing equations for a simplified FSI formulated by ALE as

$$\begin{aligned}(\dot{\mathbf{x}}^f, \mathbf{w}^f) + \underline{\mathcal{Q}}^f_{\Omega^f}\left(\mathbf{x}^f, \mathbf{w}^f\right) + \underline{\mathcal{Q}}^f_{\Gamma}\left(\mathbf{x}^f, \mathbf{w}^f\right) = \underline{\mathcal{F}}^f(\mathbf{w}^f),\\ (\dot{\mathbf{x}}^s, \mathbf{w}^s) + \underline{\mathcal{Q}}^s_{\Omega^s}\left(\mathbf{x}^s, \mathbf{w}^s\right) + \underline{\mathcal{Q}}^s_{\Gamma}\left(\mathbf{x}^s, \mathbf{w}^s\right) = \underline{\mathcal{F}}^s(\mathbf{w}^s),\end{aligned} \tag{23}$$

where $\mathbf{x}$ is a collection of all field values (geometry displacement $\mathbf{d}$, velocity $\mathbf{u}$, fluid pressure p), $\mathbf{w}$ is the test function, the superscripts s and f denote solid and fluid, respectively.

Assume a constraint $\mathbf{x}^f\big|_\Gamma = \mathbf{x}^s|_\Gamma$, and the interface terms in 23 can be replaced by $\underline{\mathcal{Q}}^f_{\Gamma}\left(\mathbf{x}^f, \mathbf{w}^f\right) = \underline{\mathcal{Q}}^f_{\Gamma}\left(\mathbf{x}^s, \mathbf{w}^f\right)$ and $\underline{\mathcal{Q}}^s_{\Gamma}\left(\mathbf{x}^s, \mathbf{w}^s\right) = \underline{\mathcal{Q}}^s_{\Gamma}\left(\mathbf{x}^f, \mathbf{w}^s\right)$, and the coupled system is rewritten as

$$\begin{aligned}(\dot{\mathbf{x}}^f, \mathbf{w}^f) + \underline{\mathcal{Q}}^f_{\Omega^f}\left(\mathbf{x}^f, \mathbf{w}^f\right) + \underline{\mathcal{Q}}^f_{\Gamma}\left(\mathbf{x}^s, \mathbf{w}^f\right) = \underline{\mathcal{F}}^f(\mathbf{w}^f),\\ (\dot{\mathbf{x}}^s, \mathbf{w}^s) + \underline{\mathcal{Q}}^s_{\Omega^s}\left(\mathbf{x}^s, \mathbf{w}^s\right) + \underline{\mathcal{Q}}^s_{\Gamma}\left(\mathbf{x}^f, \mathbf{w}^s\right) = \underline{\mathcal{F}}^s(\mathbf{w}^s),\end{aligned} \tag{24}$$

where the structural variable $\mathbf{x}^s$ acts as a term in the fluid ROM, and vice versa for $\mathbf{x}^f$.

The fluid and solid ROMs are assembled into a coupled system. The same treatments can be applied to other constraints if necessary. Readers can turn to the articles for applications in aeroelasticity (i.e., Euler equations and linear/nonlinear elastic structure) [267, 268] and the Cardiovascular System (i.e., incompressible Navier-Stokes flow and linear elasticity) [268].

We highlight several recent contributions from Nonino et al. [264, 269] that focus on transport-dominated nonlinear FSI problems and ROMs. The authors successfully addressed a crucial issue in nonlinear reduced systems, namely, the slow decay of the Kolmogorov n-width. This issue was resolved by a method called *transport maps*. They solved FSI systems adopting an ALE formulation like 24. Moreover, interface constraints can be weakly enforced by Lagrange multipliers (similar to Eq. 8). Assume the geometry displacements $\mathbf{d}$ are equal on an interface $\mathbf{d}^f = \mathbf{d}^s$. Then the additional equations are expressed as,

$$\int_\Gamma \zeta_i \left(\mathbf{d}^f - \mathbf{d}^s\right) = 0 \quad \forall \zeta_i \in \mathbb{L},$$

where $\mathbb{L}$ is a functional space for the Lagrange multipliers ζ_i.

Be aware that *bifurcating phenomena* may occur in FSI problems, which indicate non-unique stable solutions as parameters are varied [270]. Khamlich et al. studied a special bifurcating behavior (known as the Coandă effect) in a FSI problem comprised of incompressible Navier-Stokes flow and a linear elastic solid [9, 271]. In their simulations, the FOM coupling is achieved through the ALE framework, and interface constraints are enforced using Lagrange multipliers (similar to those employed by Nonino et al.). More descriptions about model order reduction for bifurcation problems can be found in [272–275].

4.2.7.2 Mesh Motion in FSI Using Interpolation The aforementioned methods (i.e., the ALE formulations) require solving a subsystem of equations to estimate the motion of the fluid mesh in cases of unsteady FSI phenomena. According to the references observed, geometric reduction of mesh morphing can also be achieved using interpolation. Control points in the structural interfaces can be used as input to predict the fluid grids at each time step.

Forti and Rozza [276] adopted RBF to create the interpolants. As the cost of RBF interpolation increases significantly with the number of control points, they proposed a greedy procedure to minimize the number of control points. The basic idea is to select surface points that can be used to accurately approximate the eigenmodes of the solid. They tested the adaptive approach in two cases: (i) modal analysis of a commercial aircraft; (ii) external viscous fluid flow past a deformable rectangular obstacle.

Apart from RBF, the *IDW* formulation can also be adopted. Witteveen et al. exploited IDW for high-fidelity FSI with a 2D airfoil and a 3D wing [115]. They demonstrated that IDW is comparable in accuracy to RBF, while reducing computational costs by around 100 times [115].

D'Amario incorporated IDW and ROMs in his thesis to simulate typical FSI phenomena: the fluid flow around an aircraft wing and a ship hull [117]. However, the IDW approach is not practical for complex geometries consisting of a large number of points. To overcome this limitation, D'Amario and Ballarin et al. proposed the SIDW method, also called Reduced IDW [117, 118]. This algorithm optimally extracts a subset of control points based on a geometric criterion, thereby reducing the cost of interpolation. Furthermore, additional constraints can be enforced to enhance SIDW and exploited the enhanced framework to analyze three benchmark cases, including the structural deformation of a wing, and fluid mesh motion around a wing and a rotating hull.

Lastly, we highly recommend the comprehensive investigation presented in Forti's thesis [114], in which RBF, IDW, and FFD are employed to compute mesh motion when constructing ROMs for FSI. Additionally, the author proposed two novel strategies for improving RBF and FFD. The adaptive section of RBF points is integrated as a hyper-reduction.

Domain decomposition is incorporated with FFD to enable different control point refinement for different locations. The performance of these techniques is compared with respect to FSI phenomena governed by incompressible Navier-Stokes equations and linear elasticity: (i) an external laminar fluid flow past an elastic obstacle and (ii) an internal flow in a cylindrical deformable vessel. In conclusion, he emphasized the significant potential of RBF in terms of computational cost and accuracy.

4.2.7.3 Embedded Boundary Method for FSI

Geometric parameterization analysis typically involves applying transformations to reference domains or remeshing. This section introduces a so-called *Embedded Boundary Method* (EBM) as an alternative approach. It incorporates the concept of domain decomposition to parameterize complex shapes and FSI problems. An important step of the framework is "extracting" embedded elements from a mesh, which can be considered a type of spatial partition and is therefore presented here.

The introduction consists of three parts: decomposition, FOM formulation, and snapshots and the ROM framework. See more in [277–280].

The decomposition and notation of the EBM can be summarized as follows (see also in Fig. 22.): (i) discretize a domain $\mathcal{B}$ that embeds the solid $\mathcal{D}$; (ii) generate a background mesh $\mathcal{B}_h$; (iii) "cut" the approximate *surrogate* geometry $\mathcal{D}_\mathcal{T}$ according to the embedded mesh $\mathcal{T}_h$ to approximate the true geometry. The elements intersecting the boundary $\partial\mathcal{D}$ denote the "cut" elements G_h. Note that the subdomain $\mathcal{D}$ can be deformed and relocated in cases of parameterization.

After decomposition, the EBM weak formulation of the concerned problem is imposed in $\mathcal{D}_\mathcal{T}$ or $\mathcal{T}_h$ as finding u_d such that

$$(\dot{u}_d, w) + \underline{\mathcal{Q}}\,(u_d, w) = \underline{\mathcal{F}}(w) \quad \text{in } \mathcal{D}_\mathcal{T}(\text{or } \mathcal{T}_h),$$

where u_d is the solution in $\mathcal{D}_\mathcal{T}$, and w is the test function. $\underline{\mathcal{Q}}$ is the EBM operator consisting of penalty terms. The exact expressions of $\underline{\mathcal{Q}}$ and $\underline{\mathcal{F}}$ for the Poisson equation is shown in Appendix B.4.

Besides, to better capture the real boundary, the EBM also incorporates additional integral terms for the "cut" elements G_h. The details of the EBM formulation for ROMs can be found in [281, 282].

The values in the rest region, $u_b$in $\mathcal{B}\setminus\mathcal{D}_\mathcal{T}$, can be obtained from two strategies, namely

$$(1)\ u_b = 0\text{in } \mathcal{B}\setminus\mathcal{D}_\mathcal{T} \quad \text{and} \quad (2)\begin{cases} -\Delta u_b = 0, & \text{in } \mathcal{B}\setminus\mathcal{D}_\mathcal{T}, \\ \quad u_b = u_d|_{\partial\mathcal{D}_\mathcal{T}}, & \text{on } \partial\mathcal{D}_\mathcal{T}, \\ \quad u_b = 0, & \text{in } \partial\mathcal{B}, \end{cases} \tag{25}$$

where (1) is called a *natural smooth extension* and (2) is a *harmonic extension*.

If the shape of $\mathcal{D}$ is parameterized, a set of FOM simulations is performed for each sample. It is obvious that all calculations are performed in the same background mesh with different $\mathcal{D}_\mathcal{T}$. Then, the solutions in $\mathcal{B}$ are collected as snapshots into the matrix $\underline{\underline{S}} = [u_1, \ldots, u_{N_\mu}]$, in which each field is given by $u_i = [u_d(\boldsymbol{\mu}_i)\ u_b(\boldsymbol{\mu}_i)]^{\mathrm{T}}$. The RB can be computed via POD for the entire computational domain. Consequently, Galerkin/least-squares projection formulations can be applied to construct a ROM.

The high-fidelity EBM has been used for analyzing fluid dynamics and FSI [277]. An early work of the EBM-based ROM was carried out by Balajewicz and Farhat [283]. The authors formulated a ROM employing the least-squares projection and tested it with three FSI problems: (i) one-dimensional FSI based on the viscous Burgers' equation; (ii) a compressible viscous flow around a heaving rigid cylindrical body; (iii) a viscous flow inside a square cavity with a rotating ellipsoidal body.

Karatzas et al. have published a sequence of studies employing the *shifted boundary* method [282, 284, 285] and *cut element FEM* [281, 286], and the POD-Galerkin ROM. The shifted boundary method adopts additional terms to account for the distance between the real boundary and the embedded boundary. The latter approach involves additional integral treatment for cutting elements intersecting with the interface to better represent real boundaries (e.g., G_h in Fig. 22).

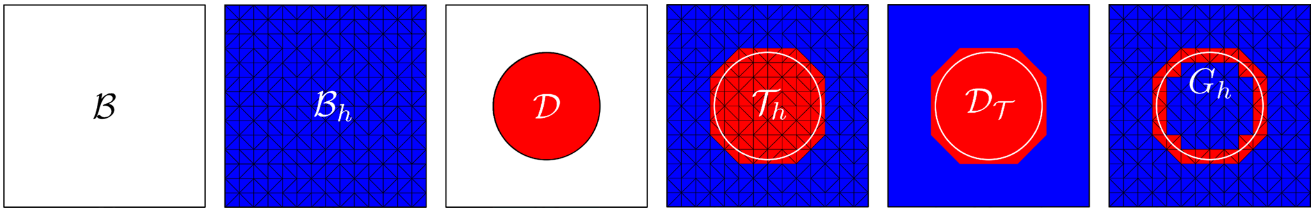


Fig. 22 Sketch and definition of the embedded boundary method. From left to right: the background geometry $\mathcal{B}$, the background mesh $\mathcal{B}_h$, the solid region $\mathcal{D}$, the *embedded* mesh $\mathcal{T}_h$, the *surrogate* geometry $\mathcal{D}_\mathcal{T}$, and the "cut" elements G_h. The location and shape of $\mathcal{D}$ can be parameterized. Figures redrawn based on [281]

The two techniques have been verified with various applications. The shifted boundary ROMs are used to analyze a heat conduction [278], a Stokes flow around an embedded circular cylinder and more complex obstacles [282], a steady incompressible Navier-Stokes flow past an embedded rectangular domain [284], a fourth-order nonlinear Cahn-Hilliard system [286], and shallow water hyperbolic equations [285]. The cut element FEM incorporating with a ROM for the Darcy flow pressure model and the Steady Stokes problem can be found in [281].

Another method that can be classified into the group is called *unfitted finite element method* [136, 137, 287, 288]. The method also incorporates the concept of embedded interfaces, and the interface conditions are weakly imposed using the Nitsche's method (see Sect. 4.2.4 for a brief introduction to Nitsche's method). The high fidelity unfitted FEM for elliptic problems is presented in [287]. To resolve the instability problem due to "cutting" small elements, an improved version is proposed in [288].

Chasapi et al. applied the method to construct ROMs for a parameterized Poisson and linear elasticity problems [136]. The authors also integrated parameter clustering method to generate local RBs regarding the parameter space and hyper-reduction technique to reduce the computational cost of the resulting ROMs. A further extension of this framework can be found in [137], where the authors applied the method to fluid dynamics problems, including incompressible Stokes and incompressible Navier-Stokes equations. The results show that the method can achieve good accuracy and efficiency for both linear and nonlinear problems. In the study, the authors also compared the performance of several reduction techniques for computing local RBs in the parameter domain and the capability of the supremizer enrichment approach (see more in [194]) to enhance the stability of the resulting ROMs.

4.2.7.4 Complex Multiphysics The coupled formulations 24 can be extended to multiphysics problems. Manzoni et al. constructed ROMs for cardiac electrophysiology and mechanics [289, 290]. We recommend [12] for a general description and [99] for applications about coupled multiphysics ROMs.

Lagrange multipliers are also applied to couple multiphysics phenomena. Corigliano et al. employed the technique to analyze nonlinear elastic-plastic structural dynamics and the electro-mechanical system [291–293].

Note that the investigations mentioned above require a two-way exchange between the models. However, in some scenarios, the interaction is one-way. For example, given a two-equation problem, one has a fixed interface condition (maybe homogeneous), and the other subproblem needs interface quantities as inputs. Zappon et al. presented both steady and unsteady one-way coupling for second-order elliptic problems [39, 294]. The authors succeeded in coupling non-conformal interfaces mesh at both the FOM and ROM stages, using a so-called *INTERNODES* method [295, 296].

4.3 Iterative

We will now continue to address the second family of coupling methodologies: *iterative* procedures. The methods have been widely used in high-fidelity numerical analysis. They can couple sub-problems based on any discretization techniques (e.g., finite differences, finite element, finite volume, and spectral element methods) [53]. The framework can also be adapted to an assembly of local ROMs regarding either overlapping or non-overlapping partitions.

The structure of the methods can be divided into two stages: (i) in each iteration, each sub-problem is solved independently, imposing data transferred from adjacent partitions as BCs or, alternatively, known global boundaries; (ii) iterations are performed considering the whole system until discontinuities or residuals satisfy a predefined threshold.

The implementations of the most widely used iterative algorithms, the Schwarz method and its variations, are described in sequence. Besides, compared to a single continuum, the coupling of multiphysics phenomena is more complex and requires more interface constraints. A separate section on the topic is presented at the end.

4.3.1 Schwarz Method

The *Schwarz method* (also referred to as *Schwarz alternating method*) is a classic iterative domain decomposition method formulated by H. A. Schwarz around 1870. It can be applied to spatial partitions with overlaps. Denoting $\Omega = \cup_{m=1}^{N_\Omega} \Omega_m$, for two overlapping parts Ω_m and Ω_n, their interfaces are defined as $\Gamma_{[mn]} = \partial\Omega_m \cap \Omega_n$ and $\Gamma_{[nm]} = \partial\Omega_n \cap \Omega_m$. The two adjoining sub-problems and their boundary and initial conditions can be formulated as

$$
\begin{cases}
\left(\dot{u}_m^{(k)}, w_m\right) + \underline{\mathcal{Q}}\left(u_m^{(k)}, w_m\right) = \underline{\mathcal{F}}(w_m) & \text{in } \Omega_m, \\
u_m^{(k)} = u_n^{(k-1)} & \text{on } \Gamma_{[mn]}, \\
u_m^{(k)} = g_D & \text{on } \partial\Omega_m \cap \Gamma_D, \\
\dfrac{\partial u_m^{(k)}}{\partial \mathbf{n}} = \mathbf{g}_N & \text{on } \partial\Omega_m \cap \Gamma_N,
\end{cases}
$$

$$
\begin{cases}
\left(\dot{u}_n^{(k)}, w_n\right) + \underline{\mathcal{Q}}\left(u_n^{(k)}, w_n\right) = \underline{\mathcal{F}}(w_n) & \text{in } \Omega_n, \\
u_n^{(k)} = \begin{cases} u_m^{(k-1)} \text{or} \\ u_m^{(k)} \end{cases} & \text{on } \Gamma_{[mn]}, \\
u_n^{(k)} = g_D & \text{on } \partial\Omega_n \cap \Gamma_D, \\
\dfrac{\partial u_n^{(k)}}{\partial \mathbf{n}} = \mathbf{g}_N & \text{on } \partial\Omega_n \cap \Gamma_N,
\end{cases}
\tag{26}
$$

where the superscript (k) indicates iteration steps, Γ_D and Γ_N are global boundaries. Depending on the iteration BCs in sub-problem 26, the method is called *multiplicative Schwarz* when choosing $u_m^{(k)}$, or *additive Schwarz* when the selection is $u_m^{(k-1)}$.

It has been proven that the Schwarz method always converges for second-order elliptic equations, with a rate that increases as the measure of the overlapping region increases [53]. Additionally, relaxation can be utilized to improve the convergence, which modifies the interface condition to be $u_m^{(k)} = \beta u_n^{(k-1)} + (1-\beta)\, u_m^{(k-1)}$, with β as the relaxation factor.

Each sub-problem is approximated separately. Thus, local ROMs can be directly constructed using Galerkin projection, resulting in a formulation expressed in terms of Eq. 4.

The early stage of the iterative scheme is addressed as a hybrid technique to couple FOMs and ROMs. Buffoni et al. [297] applied the multiplicative Schwarz method to combine a ROM and a FOM for two non-linear problems, including a Laplace equation with non-linear boundary conditions and compressible Euler equations. Cinquegrana et al. [35] analyzed the aerodynamic flow field around a 2D airfoil geometry governed by the Reynolds Average Navier-Stokes equations with the multiplicative Schwarz procedure. The region surrounding the airfoil is resolved by a high-fidelity model, and the rest is approximated with a ROM. Song and Rui [298] utilized it to analyze the convection-dominated diffusion equation for a geometry divided with overlaps. Localized global RBs were computed to approximate sub-problems.

In general, the Schwarz algorithm is utilized for divisions with overlaps. Nevertheless, a recent publication from Tezaur et al. [19] demonstrates the possibility of obtaining a stable and accurate coupled model for non-overlapping cases. Their approach is evaluated by three challenging 2D nonlinear hyperbolic problems: the shallow water equations, Burgers' equation, and the compressible Euler equations. They employed a hyper-reduction technique, which samples a few elements on the interfaces to exchange local BCs. Thus, they also assessed the effect of the boundary sampling approach.

A novelty in the frame of Schwarz iteration can be seen from the study of Discacciati et al. [55]. They incorporate the overlapping multiplicative Schwarz method for simulations of parametric elliptic problems via *Proper Generalized Decomposition* (PGD). Instead of reduction by POD, the PGD is employed to approximate high-fidelity solutions.

The basic ideology of PGD is that the solution with respect to several parameters can be approximated by a sum of separable functions. For example, given a parametric problem, if the parameter vector $\boldsymbol{\mu}$ has two subsets, namely, $\boldsymbol{\mu} = [\boldsymbol{\mu}_1 \; \boldsymbol{\mu}_2]^{\mathrm{T}}$, the solution $u(\mathbf{x}; \boldsymbol{\mu})$ is then expanded by PGD as

$$u(\mathbf{x}; \boldsymbol{\mu}) = u(\mathbf{x}; \boldsymbol{\mu}_1, \boldsymbol{\mu}_2) = \sum_{i=1}^{N^{\mathrm{PGD}}} X_i(\mathbf{x}) M_i(\boldsymbol{\mu}_1) K_i(\boldsymbol{\mu}_2).$$

An example of the PGD expansion for Once the PGD expansions are available, they can be integrated into the weak formulation to solve local problems. And each local solution is iteratively updated employing the Schwarz approach. It is easy to check that, following the previous steps, the unknowns are condensed to the interfaces, i.e., $\Lambda_q^{\Gamma_r}(\boldsymbol{\mu}_{\mathrm{BC}})$. Thus, it can be regarded as a static condensation procedure.

The method is tested in three problems considering overlapping subdomains [55]: a two-domain parametric Poisson equation, a two-domain convection-diffusion equation, and a multi-domain thermal problem with discontinuous conductivity. Note that, for the multi-partition cases, the snapshots are collected through localized training (see Sect. 3.3.3) with arbitrary Dirichlet boundary conditions at the interfaces.

4.3.2 Variations of the Schwarz Method

The classic Schwarz method accomplishes *Dirichlet-Dirichlet* coupling between adjacent subdomains but can be extended to Neumann and Robin conditions in a non-overlapping framework. For *Dirichlet-Neumann*, the interface condition is formulated as,

$$\left.\begin{aligned} u_m^{(t)} &= u_n^{(t-1)} \\ \frac{\partial u_n^{(t)}}{\partial \mathbf{n}} &= \frac{\partial u_m^{(t)}}{\partial \mathbf{n}} \end{aligned}\right\} \quad \text{on } \Gamma_{[mn]}. \tag{27}$$

Similarly, the *Robin-Robin* iteration is written as

$$\left.\begin{aligned} \frac{\partial u_m^{(t)}}{\partial \mathbf{n}} + \gamma_m u_m^{(t)} &= \frac{\partial u_n^{(t-1)}}{\partial \mathbf{n}} + \gamma_m u_n^{(t-1)} \\ \frac{\partial u_n^{(t)}}{\partial \mathbf{n}} + \gamma_n u_n^{(t)} &= \frac{\partial u_m^{(t)}}{\partial \mathbf{n}} + \gamma_n u_m^{(t)} \end{aligned}\right\} \quad \text{on } \Gamma_{[mn]}, \tag{28}$$

where γ_m and γ_n are non-negative parameters to control the importance of each term. The choice of these parameters will affect the convergence and should be determined from a sensitivity study [299]. As indicated in [37], the Robin parameter γ is prior fixed to ensure the well-posedness of the governing equation, which typically requires $\gamma > 0$.

We start the discussion with studies that employ the Dirichlet-Neumann method. Maier and Haasdonk [300, 301] implemented it for an elliptic PDE (i.e., static heat equation with homogeneous Dirichlet boundary conditions). Separated RBs were computed for interfaces and inner regions. They managed to glue RBs belonging to two non-overlapping subdomains of distinct shapes. A posteriori error estimation concludes that the residuals of the ROM were comparable to the FOM.

The framework was then developed for incompressible Navier-Stokes equations by Reyes [302]. The authors adopted a non-overlapping decomposition. Additionally, they revealed that the performances of the Dirichlet-Neumann are comparable to the local DG ROM (see Sect. 4.2.4.).

Tezaur et al. [36] further tested its possibility for coupling two FOMs (FOM-FOM), a FOM an ad ROM (FOM-ROM), and even two ROMs (ROM-ROM). In their study, snapshots were collected from a sequence of FOM-FOM simulations. Individual RBs were computed for each subdomain. The performance of different coupling systems (FOM-ROM, ROM-ROM, etc) was evaluated on a 1D nonlinear wave propagation problem. The authors also revealed that the performance of Dirichlet-Neumann and Dirichlet-Dirichlet is comparable [303]. We also recommend readers two presentations addressed by Tezaur et al. [303, 304], in which they intended to employ the Schwarz alternating scheme to achieve the coupling of multi-scale and multi-physics problems, e.g., FSI approximated by a FOMs-ROMs system. Furthermore, they also discussed the possibility of creating a general framework to couple FOMs with intrusive and/or non-intrusive ROMs for more complex and realistic applications.

Several up-to-date publications [305, 306] have demonstrated that the Dirichlet-Neumann iteration scheme is suitable for solid mechanics, e.g., contact problems in elastodynamics.

A recent innovation by Zappon et al. [34] is utilizing the Dirichlet-Neumann to couple problems discretized with non-conforming high-fidelity mesh for different subdomains. The interpolation between the non-conforming meshes at the interface is achieved by *INTERNODES* [295, 296] method. Furthermore, hyper-reduction based on a *discrete empirical interpolation method* [169] is applied, which allows interpolating the parametric Dirichlet-Neumann interface conditions with a small amount of *magic points*. Their approach has been tested in several second-order elliptic/parabolic problems that consider two partitions.

Except for Dirichlet-Neumann, Discacciati and Hesthaven integrated the Robin-Robin approach to analyze a series of parametrized nonlinear problems, including Advection-diffusion-reaction coupling, time-dependent diffusion equation in multiple domains, etc [37]. The authors also proposed a localized training procedure (see Sect. 3.3.3) by parametrizing Robin conditions. Snapshots and RBs are constrained in archetype components.

The *Neumann-Neumann* iterative algorithm differs slightly from the methods above. It requires solving additional correction problems. Due to the lack of applications for local ROMs coupling (to the best of our knowledge), it is not illustrated here. Its formulation is presented in [53].

4.3.3 Multiphysics Problems

For the studies mentioned above, a pair of conditions is exchanged among adjacent subdomains. However, for multiphysics problems, more constraints must be enforced at the interfaces.

Since FSI is the most typical multiphysics problem, we describe it first. For a problem combining incompressible Navier-Stokes flow and elasticity, three interface conditions are required to guarantee geometric consistency, velocity continuity, and the balance of normal forces [266]. They are defined on Γ as

$$\begin{aligned} \mathbf{d}^f - \mathbf{d}^s &= 0, \\ \mathbf{u}^f - \mathbf{u}^s &= 0, \\ \sigma^f \cdot \mathbf{n}^f - \sigma^s \cdot \mathbf{n}^s &= 0, \end{aligned}$$

where superscripts f and s indicate fluid and solid respectively, $\mathbf{d}$ is the displacement of the geometry, $\mathbf{u}$ is the velocity ($\mathbf{u}^s = \frac{\partial \mathbf{d}^s}{\partial t}$), $\mathbf{n}$ is the normal vector and $\sigma \cdot \mathbf{n}$ is the normal force on the interfaces.

Note that although more variables are considered in multi-physics scenarios, iterative schemes still support the coupling. The stepping interface conditions are formulated as those in Eqs. 26, 27, and 28. This coupling procedure can be applied to both FOM and ROM levels. Some applications are presented in [307–309].

For other multiphysics problems, different interface constraints should be enforced, but the iterative algorithm remains the same. In this context, we found two studies particularly interesting regarding the implementation of multiphysics: (i) linear thermo-poroelasticity [310]; (ii) nuclear reactor thermal-hydraulics (Navier-Stokes flow with energy balance) coupled with neutronics (scalar diffusion equation) [311].

5 Data-Driven Techniques

The rapid development of *machine learning* techniques and computational power has recently encouraged the research community to incorporate pure data-driven methods to construct ROMs. These approaches are fundamentally non-intrusive, as they operate directly on the high-fidelity solutions without requiring any modification of the underlying governing equations. A review regarding the topic was published in 2021 [51]. However, various studies have emerged in the past five years, so we present an overview including the most recent advances in this area. Additionally, several relevant references reviewed in [51] are included for completeness.

In the following sections, we will provide the basic preliminaries for constructing data-driven models. Then, the strategies for combining domain decomposition and ROMs will be explained with respect to each technique. In our opinion, the classification of the non-intrusive approaches is not as straightforward as the intrusive ones. This difficulty arises from the variety of machine learning frameworks and the flexibility of incorporating different techniques to couple the local ROMs. Thus, considering their fundamental principles, we have created four categories to roughly classify them. Those are: (i) Schwarz-based iteration; (ii) interpolation algorithm; (iii) optimization-based technique; (iv) *Physics-Informed Neural Network* (PINN) based methods.

The classification of the data-driven techniques is illustrated in Fig. 23.

It is worth noting that the domain decomposition and basis construction strategies illustrated in Sect. 3.1 also support the narrations hereafter. Additionally, the coupling strategies of the aforementioned intrusive procedures may be incorporated into the non-intrusive systems.

5.1 Preliminaries

The fundamental ideology of the data-driven strategy is building a surrogate model that correlates inputs and outputs. The model is trained and validated on sampled parameters and corresponding results. Once available, it can make predictions for unknown parameters with a cost several orders of magnitude lower than FOM calculations. The surrogate model can be built using various techniques, including conventional regression algorithms and neural networks, especially the more advanced PINNs. They will be briefly explained in the paragraphs that follow.

5.1.1 Regression Algorithms

Regression processes within the field of machine learning are diverse. Least squares or linear regression can be utilized for simple scenarios. For more complex conditions, several advanced methods are adopted, such as *RBF interpolation* [111] and *gaussian process regression* [312], as well as *Artificial Neural Network* (ANN) [313].

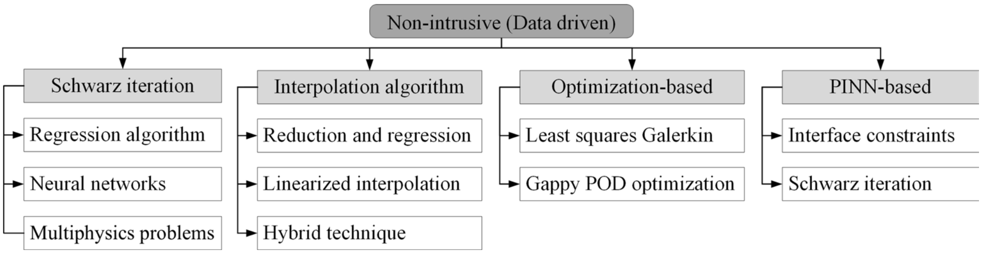
Suppose that $\{u(\boldsymbol{\mu}_j)\}_{j=1}^{N_\mu}$ is a set of general fields to be predicted, where $\boldsymbol{\mu}_j$ denotes the corresponding parameter vector. The construction of a regression model consists of *training* and *testing* stages. We denote the surrogate operator by $\underline{\mathcal{Z}}$, which maps from the parameter space to the solution space, $\underline{\mathcal{Z}} : \mathbb{R}^{N_\mu} \to \mathbb{R}^{N_u}$. Using the observed parameter-solution pairs, the model is trained so that the predicted field matches the target field, $u(\boldsymbol{\mu}) \approx \underline{\mathcal{Z}}(\boldsymbol{\mu})$. Then, for a new parameter $\boldsymbol{\mu}'$, the corresponding prediction is given by $u(\boldsymbol{\mu}') \approx \underline{\mathcal{Z}}(\boldsymbol{\mu}')$.

Moreover, dimensionality reduction techniques like POD (i.e., $u \approx \sum_{i=1}^{N_{\mathrm{RB}}} \alpha_i v_i$) can be involved in the procedure. Instead of regressing whole snapshots, the surrogate operator can also be used to predict POD coefficients, namely, $\alpha_i'(\boldsymbol{\mu}) \approx \underline{\mathcal{Z}}_i(\boldsymbol{\mu}')$. The reconstructed field $u'(\boldsymbol{\mu}) \approx \sum_{i=1}^{N_{\mathrm{RB}}} \alpha_i'(\boldsymbol{\mu}) v_i$ [9, 17]. This approach is called *POD interpolation* (PODI).

The dimensionality reduction can also be accomplished by applying neural networks, like the *autoencoder* [15]. The technique can generate a vector to represent high-dimensional data, which is referred to as *latent variables* (also known as a latent space). More details about the autoencoder can be found in Appendix B.5.

Note that, due to the variety of regression processes, their formulations cannot be explained in detail here. We suggest that readers consult the books [15, 111, 312, 314] for more detailed explanations.

Fig. 23 The classification of data-driven techniques



5.1.2 Neural Network

Recently, ANNs have been widely incorporated to construct regression frameworks for complex and nonlinear situations [17]. Considering its relevance to the following narration, we briefly discuss the basic idea of the ANN. Fig. 24 illustrates a simple example of a network architecture. Parameters $\boldsymbol{\mu}$ are regarded as inputs, namely the *input layer*. Target results are extracted from the *output layer*. The quantities pass through several *hidden layers* to build a surrogate for inputs and outputs.

The aim of a neural network in the training stage is to reduce the *loss function*, which is a measure of the discrepancy between the network's predictions and the actual target values. Assuming the target is a scalar field $u(\boldsymbol{\mu})$ and the model's prediction is $\underline{\mathcal{Z}}(\boldsymbol{\mu})$, the *Mean Squared Error* (MSE) of the network is defined as

$$\mathcal{E} = \frac{1}{N_u}\|u(\boldsymbol{\mu}) - \underline{\mathcal{Z}}(\boldsymbol{\mu})\|_2^2, \tag{29}$$

where N_u denotes the dimension of $u(\boldsymbol{\mu})$.

The minimization is achieved through the propagation of values across the neurons. For each neuron, the input x and output y are related by

$$y = \underline{\sigma}(wx + b), \tag{30}$$

where $\underline{\sigma}$ is the *activation function*, and w and b denote the *weight* and *bias*, respectively. $\underline{\sigma}$ is a nonlinear function that introduces nonlinearity into the network, which should be chosen based on the specific problem and data characteristics. w and b are tuned to minimize the loss, i.e., the training stage. More in-depth discussions regarding the implementations are not presented here, but can be found in [313].

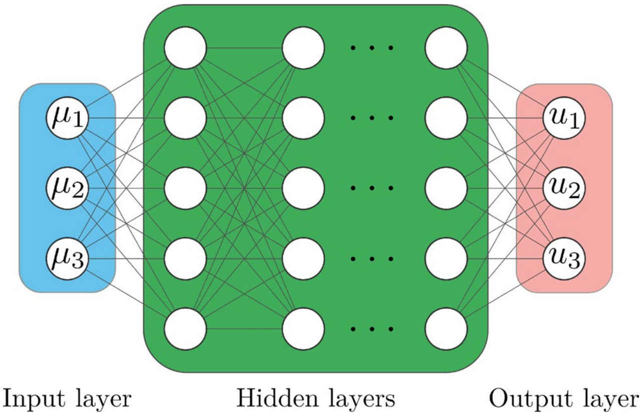


Fig. 24 A sketch of neural network architecture. The inputs are the components of the parameter vector $\boldsymbol{\mu}$, namely μ_1, μ_2, and μ_3, and the outputs are the target results, u_1, u_2, and u_3. The quantities pass through several hidden layers to build a surrogate for inputs and outputs

The minimization is achieved through the propagation of values across the neurons. For each neuron, the input x and output y are related by

$$y = \underline{\sigma}(wx + b), \tag{31}$$

where $\underline{\sigma}$ is the activation function, and w and b denote the weight and bias, respectively.

There are various types of neural networks, such as *recurrent neural networks* and *convolutional neural networks*. Their architectures are different from the one shown in Fig. 24. More details can be found in Appendix B.5.

5.1.3 Physics-Informed Neural Network

A PINN is a type of regression model that involves knowledge of physical phenomena (i.e., PDEs) within the framework of neural networks. It was proposed by Karniadakis et al. in 2017 [315, 316], and has recently been adapted for domain decomposition problems. The following paragraphs provide a brief introduction to the PINN. Its implementations for local ROMs will be presented in Sect. 5.5 and 5.2.

In contrast to the typical networks that aim to match input and output (see Eq. 29), the loss function of a PINN is intrinsically designed to integrate governing PDEs, boundary, and initial conditions. Namely, the MSE of a PINN is given by

$$\mathcal{E} = w_f\mathcal{E}_f + w_b\mathcal{E}_b, \tag{32}$$

where $\mathcal{E}_f$ and $\mathcal{E}_b$ are the PDE loss using predicted solutions and the error results from the boundary and initial conditions, respectively. The two are weighted by factors w_f and w_b to comprise the total MSE. Remark that $\mathcal{E}_f$ can be either residuals of the governing equations or an optimal formulation of the model problem. Properties of the MSE loss (Eq. 32) guarantee that the PINN's predictions satisfy both the governing equations and physical constraints.

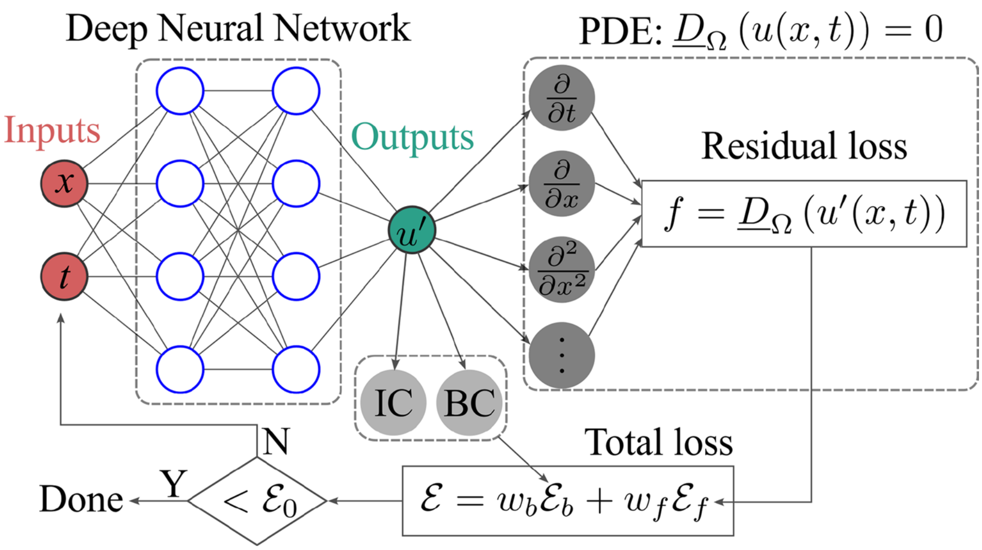
The architecture of a PINN is depicted in Fig. 25, which consists of a distinct structure compared to standard networks. The inputs are spatial coordinates $\mathbf{x}$ and temporal coordinate t, and the output is the solution $u(\mathbf{x}, t)$. Suppose a PDE defined in the computational domain Ω is given by

$$\underline{\mathcal{D}}_\Omega\big(u(\mathbf{x}, t)\big) = 0,$$

and the PINN prediction u', the residual of the PDE, which is also known as the physics-informed loss, is given by

$$\mathcal{E}_f = \frac{1}{N_f}\|\underline{\mathcal{D}}_\Omega\left(u'(\mathbf{x}, t)\right)\|_{2,\Omega}^2,$$

Fig. 25 Architecture of a PINN [317]. The inputs are coordinates $\mathbf{x}$ in the computational domain and time t. The governing equations are represented by $\underline{\mathcal{D}}_{\Omega}\left(u(\mathbf{x},t)\right)=0$. The outputs are physical variables $u'(\mathbf{x},t)$ over the spatial and temporal domains. The total loss $\mathcal{E}$ consists of two parts: the PDEs loss $\mathcal{E}_f$ and the error with respect to the constraints (initial and boundary conditions) $\mathcal{E}_b$. They are weighted by factors w_f and w_b. The *Deep neural Network* is trained to minimize $\mathcal{E}$. The algorithm ends when $\mathcal{E}$ is smaller than a predefined threshold $\mathcal{E}_0$



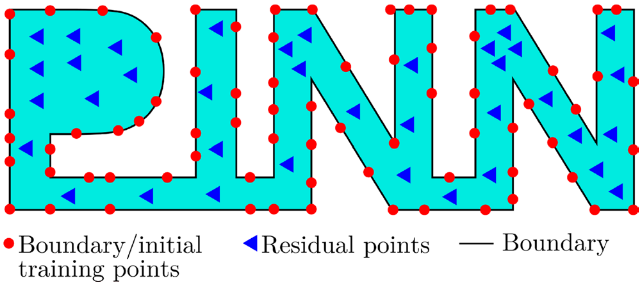


Fig. 26 PINN training data points for a domain to compute loss functions. Figure redrawn based on [318]

where $\mathbf{x}$ and t are collocation points in the computational domain for computing the PDE loss, and N_f denotes the number of data points in the interior of the computational domain Ω.

The loss regarding the boundary and initial conditions is computed by

$$\mathcal{E}_b=\frac{1}{N_b}\|u(\mathbf{x},t)-u'(\mathbf{x},t)\|^2_{2,\partial\Omega}+\frac{1}{N_f}\|u(\mathbf{x},0)-u'(\mathbf{x},0)\|^2_{2,\Omega},$$

where $\partial\Omega$ denotes the boundary of Ω, N_b and N_f are the number of data points on the boundary and in the interior, respectively, for computing the losses.

The inputs $\mathbf{x}$ and outputs $u(\mathbf{x})$ of the PINN framework are confined in a set of pre-selected data points as shown in Fig. 26. The points are divided into groups for the computation of boundary/initial losses $\mathcal{E}_b$ and the residual loss $\mathcal{E}_f$. The PINN is trained only using the points during the offline stage. Once converged, it can be utilized to predict solutions at any location and time step within the computational domain. Thus, the selections of training points significantly affect the accuracy of a PINN. Details about those aspects can be seen in the specific review articles [319, 320].

Note that the inputs and outputs of a PINN are interconnected through a series of neurons as defined in Eq. 31. This architecture enables the computation of derivatives using the chain rule, for example, $\partial u(\mathbf{x},t)/\partial\mathbf{x}$ and $\partial u(\mathbf{x},t)/\partial t$, as well as higher-order derivatives. This capability forms the mathematical foundation for evaluating the residual loss term $\mathcal{E}_f$. This approach is known as *automatic differentiation* in the framework of computational algebra [321].

Unlike traditional ROMs, PINNs can solve PDEs even without requiring data. The approximations can be obtained purely by satisfying governing equations and constraints, as shown in Fig. 25. However, when FOM results are available at specific locations, the standard data MSE (Eq. 29) can be integrated into the total PINN loss (Eq. 32), resulting in

$$\mathcal{E}=w_f\mathcal{E}_f+w_b\mathcal{E}_b+w_d\mathcal{E}_d,$$

where $\mathcal{E}_d=\frac{1}{N_u}\|u(\mathbf{x},t)-\underline{\mathcal{Z}}(\mathbf{x},t)\|^2_2$ is the data MSE, and w_d is the corresponding weight. The data loss term can enhance the accuracy of a PINN when FOM results are available, especially for complex problems.

The inputs and outputs of PINNs depend on the problem's configuration. Recently, PINNs have been intensively employed to solve both forward and inverse problems [318, 322–324]. In forward problems, PINNs approximate solutions of PDEs with fully specified parameters, boundary conditions, and initial conditions. The network inputs typically consist of spatial coordinates $\mathbf{x}$, temporal coordinates t, and parameters $\boldsymbol{\mu}$, while the output is the solution field $u(\mathbf{x},t;\boldsymbol{\mu})$ over the computational domain. In contrast, inverse systems aim to identify unknown parameters or conditions from observed data. Hence, a data loss term that enforces consistency with observed solutions is included. The inverse PINN uses the same inputs as the forward PINN, but provides the solution field and additional quantities of interest as outputs. More references for their implementations are presented in Sect. 5.5.

It is important to emphasize that while PINNs involve knowledge of the governing PDEs, their principle differs significantly from traditional intrusive methods. Recall that intrusive techniques numerically solve reduced algebraic systems derived from the PDEs. In contrast, the PINN framework replaces conventional solvers with neural networks that learn the solution operator. Since PINNs do not explicitly solve the PDE system through algebraic manipulation, they are appropriately classified as non-intrusive ROMs.

5.2 Schwarz-Based Techniques

We now start reviewing the Schwarz-based techniques in the category of data-driven methods. We will give a general description and then review the implementations in the references. Note that not only Dirichlet but also Neumann conditions are discussed in this subgroup.

The approaches that will be presented are almost the same as the Schwarz method and its variations. Instead of solving intrusive ROMs to update solutions in subdomains, non-intrusive surrogate models are built to solve each subproblem employing interface conditions and iteratively update local approximations until convergence is achieved.

5.2.1 Formulations

We adopt multiplicative Schwarz (Eq. 26) for the explanation, but the algorithms are also suitable for other conditions presented in Sect. 4.3.

Assume two overlapping subdomains Ω_m and Ω_n, and their interfaces $\Gamma_{[mn]} = \partial\Omega_m \cap \Omega_n$ and $\Gamma_{[nm]} = \partial\Omega_n \cap \Omega_m$. The iterative process of the Schwarz method is achieved by two operators: the boundary value solver $\underline{\mathcal{S}}$ and the restriction operator $\underline{\mathcal{T}}$. They are discussed in the following paragraphs.

Suppose the other boundaries of Ω_m are fixed, the local solution u_m is obtained by solving the local problem with the boundary condition $u_m|_{\Gamma_{[mn]}}$. We denote the boundary value solver for the local problem in Ω_m as $\underline{\mathcal{S}}_m$, which is defined as $u_m = \underline{\mathcal{S}}_m \left(u_m|_{\Gamma_{[mn]}} \right)$.

As mentioned in Eq. 26, the local solution u_m at step t is updated by the boundary condition $u_m^{(t)}\Big|_{\Gamma_{[mn]}} = u_n^{(t-1)}\Big|_{\Gamma_{[mn]}}$. Thus, by adopting $\underline{\mathcal{S}}_m$, the local solution u_m at step t is given by

$$u_m^{(t)} = \underline{\mathcal{S}}_m \left(u_m^{(t)}\Big|_{\Gamma_{[mn]}} \right) = \underline{\mathcal{S}}_m \left(u_n^{(t-1)}\Big|_{\Gamma_{[mn]}} \right). \tag{33}$$

To further process the solution, we need to extract the interface values $u_m^{(t)}\Big|_{\Gamma_{[nm]}}$ to update u_n at step t. This is achieved by the restriction operator $\underline{\mathcal{T}}_m$, which is defined as

$$\underline{\mathcal{T}}_m(u_m^{(t)}) := u_m^{(t)}|_{\Gamma_{[nm]}}. \tag{34}$$

Here $\underline{\mathcal{T}}_m$ maps the local solution space V_m to the appropriate trace space on the interface (e.g. $\underline{\mathcal{T}}_m : V_m \to H^{1/2}(\Gamma_{[nm]})$).

We combine Eqs. 33 and 34 to update the local solutions at step t as follows:

$$\underline{\mathcal{T}}_m(u_m^{(t)}) = \underline{\mathcal{T}}_m(\underline{\mathcal{S}}_m(u_n^{(t-1)}\Big|_{\Gamma_{[mn]}})), \quad \text{and} \quad \underline{\mathcal{T}}_n(u_n^{(t)}) = \underline{\mathcal{T}}_n(\underline{\mathcal{S}}_n(u_m^{(t)}\Big|_{\Gamma_{[nm]}})).$$

To simplify the notation, we denote $\underline{\mathcal{Z}}_m = \underline{\mathcal{T}}_m \circ \underline{\mathcal{S}}_m$ and $\underline{\mathcal{Z}}_n = \underline{\mathcal{T}}_n \circ \underline{\mathcal{S}}_n$. Then, the above formulation is rewritten as

$$u_m^{(t)}\Big|_{\Gamma_{[nm]}} = \underline{\mathcal{Z}}_m \left(u_n^{(t-1)}\Big|_{\Gamma_{[mn]}} \right), \quad \text{and} \quad u_n^{(t)}\Big|_{\Gamma_{[mn]}} = \underline{\mathcal{Z}}_n \left(u_m^{(t)}\Big|_{\Gamma_{[nm]}} \right).$$

Moreover, the relaxation strategy can be adopted to improve the convergence of the Schwarz method [52]. Thus, the final coupling for u_m and u_n at step t is given by

$$\begin{aligned} u_m^{(t)}\Big|_{\Gamma_{[nm]}} &= \underline{\mathcal{Z}}_m \left(\omega\, u_n^{(t-1)}\Big|_{\Gamma_{[mn]}} + (1-\omega)\, u_n^{(t-2)}\Big|_{\Gamma_{[mn]}} \right), \quad \text{and} \\ u_n^{(t)}\Big|_{\Gamma_{[mn]}} &= \underline{\mathcal{Z}}_n \left(\tilde{u}_m^{(t)}\Big|_{\Gamma_{[nm]}} \right) \text{ with } \tilde{u}_m^{(t)}\Big|_{\Gamma_{[nm]}} = \omega\, u_m^{(t)}\Big|_{\Gamma_{[nm]}} + (1-\omega)\, u_m^{(t-1)}\Big|_{\Gamma_{[nm]}}, \end{aligned}$$

where ω is the relaxation factor.

Remark that the dimensionality reduction techniques like POD and neural networks are generally employed beforehand. The resulting low-dimensional variables are then used to construct a surrogate operator $\underline{\mathcal{Z}}$.

5.2.2 Applications

Now, the problems turn to creating a model to approximate the operator $\underline{\mathcal{Z}}$ for each subdomain. It is clear that the construction of the surrogate model $\underline{\mathcal{Z}}$ is not unique. The following paragraphs will discuss the implementation of both regression algorithms and neural networks to construct local ROMs.

Cheng et al. used the method to study the 2D nonlinear elliptic equations [325]. They collected snapshots utilizing localized training and oversampling (see Sect. 3.3.3). The $\underline{\mathcal{Z}}$, indeed a ROM, is constructed by *two-layer neural betworks*. The authors also showed several relevant preprocessing and postprocessing, and comparisons of different configurations.

The technique can be extended to Dirichlet-Neumann conditions for non-overlapping divisions. In such a case, two surrogate models $\underline{\mathcal{Z}}^D$ and $\underline{\mathcal{Z}}^N$ are constructed to exchange Dirichlet and Neumann conditions on a shared interface. The coupling has the form of $u_m^{(t)}\Big|_{\Gamma_{[mn]}} = \underline{\mathcal{Z}}_n^D \frac{\partial u_n^{(t-1)}}{\partial \mathbf{n}}\Big|_{\Gamma_{[mn]}}$ and $\frac{\partial u_n^{(t)}}{\partial \mathbf{n}}\Big|_{\Gamma_{[mn]}} = \underline{\mathcal{Z}}_m^N u_m^{(t)}\Big|_{\Gamma_{[mn]}}$ for a two-subdomain system with $\Omega = \Omega_m \cup \Omega_n$ and $\Gamma_{[mn]} = \partial\Omega_m \cap \partial\Omega_n$.

Discacciati et al. adopted the Dirichlet-Neumann iteration (named *boundary-to-boundary mapping* approach) to analyze a series of linear and nonlinear multi-physics scenarios [326], including steady and transient nonlinear heat equations, as well as a FSI problem. The localized training (see Sect. 3.3.3) is utilized to generate datasets. POD is adopted to reduce the dimensionality of snapshots. Regression-based ROMs are built using two techniques, ANN and *Vectorial Kernel Orthogonal Greedy Algorithm*. The technical aspects of the two models are beyond the scope of this review. They are described in the original document. The method is also applied to both conforming and non-conforming high-fidelity meshes. Note that the authors also published an intrusive version of the framework, which is presented in Sect. 4.3.2 [37].

Ruan [327] used the Dirichlet-Neumann framework to couple local ROMs for a large-scale Reynolds-averaged Navier-Stokes problem. The local ROMs are constructed by the PODI technique. The modeled case is a realistic industrial scenario, flow in a wire-wrapped rod bundle geometry, which is a typical configuration in nuclear reactors. The results show that local RBs can capture the dominant flow features, and the coupling strategy can effectively exchange information between subdomains.

5.2.3 Multiphysics Phenomena

The procedure can also be applied to multiphysics problems, especially FSI phenomena.

Xiao et al. [328] utilized POD and radial basis functions to create ROMs for several test cases: (i) flow past a cylinder; (ii) 2D free-falling square in water; (iii) vortex-induced vibrations of an elastic beam. The authors coupled Navier-Stokes flow and linear elastic structural motion.

Gerdroodbary et al. [329] used POD and *convolutional neural network* to extract dominant features. Subsequently, they utilized *long short-term memory* neural network to build a surrogate model. They simulated an Abdominal Aortic Aneurysm governed by unsteady Reynolds average Navier-Stokes and elastodynamics equations. A significant conclusion can be drawn from this study: the POD-based ROM is more efficient, while the pure neural network model is more accurate.

Zhang et al. [330] incorporated an *autoencoder* for data reduction and a so-called *sparse identification of the non-linear dynamics* algorithm to identify governing equations with respect to low-dimensional variables. The innovative framework was tested in a 2D case, flow past a cylinder.

Moreover, the coupling of a ROM and a FOM can also follow the procedure. Han et al. [331] proposed a method that couples a deep neural network-based fluid ROM and a high-fidelity structural dynamic solver.

For investigations for multiphysics applications, beyond FSI, we suggest two references: (i) [332] that uses deep learning-based approaches for micro-electro-mechanical-systems; (ii) [311] that couples neutronics and thermal-hydraulics in nuclear reactor cores.

5.3 Interpolation Algorithm

The second group is for methods that employ interpolations. We will present the general principles of the interpolation-based approach, following studies that implement it in different ways.

The interpolation scheme employs iterations to couple local ROMs, and generally, the dimensionality reduction (e.g., POD) is applied to pre-process the high-fidelity data. The steps for constructing an interpolation-based ROM for domain decomposition problems are outlined in the following paragraphs.

Suppose the POD is used to compute the reduced basis of each subdomain, $u_m = \sum_i^{N_{\mathrm{RB,m}}} \alpha_{m,i} v_{m,i}$. Recall that the indices of all neighbors of Ω_m are included in a set $N_\Gamma(m) = \{n|\partial\Omega_m \cap \partial\Omega_n \neq \emptyset\}$. Then, an interpolation model $\underline{\mathcal{Z}}_m$ for the POD coefficient $\boldsymbol{\alpha}_m$ of Ω_m can be expressed as

$$\boldsymbol{\alpha}_m^{(t)} = \underline{\mathcal{Z}}_m \left(\boldsymbol{\alpha}_m^{(t-1)}, \boldsymbol{\alpha}_n^{(t-1)}, \boldsymbol{\mu}_m\right) \quad n \in N_\Gamma(m), \tag{35}$$

where $\boldsymbol{\alpha}_m = \left[\alpha_{m,1}, \ldots, \alpha_{m,N_{\mathrm{RB,m}}}\right]^{\mathrm{T}}$ is a vector containing coefficients, $\underline{\mathcal{Z}}_m$ is the interpolation model, $\boldsymbol{\mu}_m$ is a vector of local parameters for Ω_m[25], and t is the step number of iterations. The equation indicates that the subdomain approximation at iteration step (t) depends on several inputs, including its previous results in $(t-1)$ and the neighbours' results in $(t-1)$, as well as parameters $\boldsymbol{\mu}_m$.

To be clearer, an example is depicted in Fig. 27, in which the target division is Ω_5 and its neighbors are Ω_2, Ω_4, Ω_6, and Ω_8. Therefore, we have $\boldsymbol{\alpha}_5^{(t)} = \underline{\mathcal{Z}}_5 \left(\boldsymbol{\alpha}_5^{(t-1)}, \boldsymbol{\alpha}_2^{(t-1)}, \boldsymbol{\alpha}_4^{(t-1)}, \boldsymbol{\alpha}_6^{(t-1)}, \boldsymbol{\alpha}_8^{(t-1)}, \boldsymbol{\mu}_5\right)$. Also, in case a subdomain is connected to the global

---

[25] Each subdomain Ω_m can be parameterized separately. Therefore, we use $\boldsymbol{\mu}_m$ to denote the parameters for Ω_m.

boundary, the boundary conditions can be involved in the interpolation.

Be aware that the above approach can incorporate other dimensionality reduction algorithms, and various techniques can be applied to build the regression model $\underline{\mathcal{Z}}$, e.g., those introduced in Sect. 5.1. Moreover, the ways of incorporating $\underline{\mathcal{Z}}$ to build ROMs are diverse. Hence, after reviewing the references, we identify three interpolation strategies. We will provide their ideologies and applications for different equations.

5.3.1 Standard Interpolation

The first method we presented comprises the two steps above, namely, dimensionality reduction and interpolation. The studies share the same procedure, while they differ in the techniques employed in the two stages.

Xiao et al. have published several investigations exploiting the pure regression strategy [56, 334, 335]. They studied several large-scale problems governed by transient incompressible Navier-Stokes equations: (i) 2D flow past a cylinder; (ii) 2D and 3D urban street canyon test case; and (iii) 3D air flow around London South Bank University.

Although the three applications are distinct, their procedures are nearly the same. The entire geometry is decomposed similarly into non-overlapping subdomains (as Fig. 27). The global snapshots are computed and extracted for the partitions. POD is applied to calculate a set of separated dominant modes for each of them (see Sect. 3.3.2). The ROM is constructed through PODI with RBF, and a different number of POD modes is adopted for different regions to approximate the global fields. The results of the non-intrusive domain decomposition ROM are comparable to the global ROM, and verified against FOM simulations. One advantage of the local approach is that different RBs can be used to approximate each subdomain. Fewer modes are required for "less important" areas, while more are used for dominant subdomain-level variances. This maintains the accuracy of ROMs and avoids stability issues that can arise from using too many POD modes.

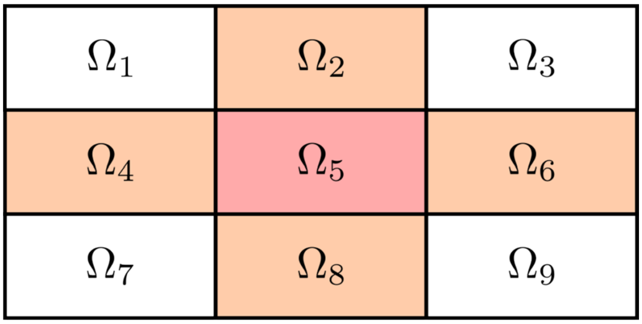


Fig. 27 A subdomain and its neighbours. Redraw based on [333]

Besides POD, Heaney et al. applied the *Autoencoder* (AE) to investigate two incompressible Navier-Stokes problems, a 2D single-phase flow past a cylinder and 3D two-phase in-pipe flow [18]. In these cases, there are no overlaps among the divisions. Four dimensionality reduction methods are used and compared in their analysis, namely POD, a *convolutional AE*, an *adversarial AE*, and a *hybrid singular value decomposition AE*. A *predictive neural network* is constructed to achieve the interpolation.

The flow past a cylinder case is modeled considering global snapshots and local RB (see Sect. 3.3.2). For the multiphase case, the tube is split axially into partitions of the same shape. Then, a set of generic local POD modes or AE latent space is utilized to reconstruct the global solutions. This is the generic decomposition and local RB construction indicated in Sect. 3.3.2. Thus, the authors managed to extend a ROM built from ten subdomains to approximate a longer pipe that consists of 100 subdomains. They concluded that the AE-based reduced techniques outperform POD, and the best among the four techniques is the convolutional AE.

5.3.2 Linearized Interpolation

A modification of the above direct interpolation algorithm is proposed by Xiao et al. [333, 336, 337] to model a huge water reservoir (e.g., artificial lake), namely *subdomain POD-trajectory-piecewise-linearization*. Given the prediction of the last step $\boldsymbol{\alpha}_m^{(t-1)}$ and parameter $\boldsymbol{\mu}_m$, the surrogate model $\underline{\mathcal{Z}}_m$ can predict a new solution using a first-order expansion around the training data point:

$$\boldsymbol{\alpha}_m^{(t)} = \boldsymbol{\alpha}_{tr}^{(t)} + \underline{\underline{E}}_{m,\boldsymbol{\alpha}_{\mathrm{tr}}}^{(t)} \left(\boldsymbol{\alpha}_m^{(t-1)} - \boldsymbol{\alpha}_{tr}^{(t-1)} \right) + \underline{\underline{G}}_{m,\boldsymbol{\mu}_{\mathrm{tr}}}^{(t)} \left(\boldsymbol{\mu}_m - \boldsymbol{\mu}_{tr} \right),$$

with

$$\underline{\underline{E}}_{m,\boldsymbol{\alpha}_{\mathrm{tr}}}^{(t)} = \frac{\partial \underline{\mathcal{Z}}_m^{(t)}}{\partial \boldsymbol{\alpha}_{tr}^{(t-1)}}, \quad \underline{\underline{G}}_{m,\boldsymbol{\mu}_{\mathrm{tr}}}^{(t)} = \frac{\partial \underline{\mathcal{Z}}_m^{(t)}}{\partial \boldsymbol{\mu}_{tr}},$$

where $(\boldsymbol{\alpha}_{tr}^{(t)}, \boldsymbol{\alpha}_{tr}^{(t-1)}, \boldsymbol{\mu}_{tr})$ belongs to the *closest* training set to $\boldsymbol{\alpha}_m^{(t)}$. The selection algorithm for closeness is presented in [338]. Note that $\underline{\mathcal{Z}}_m$ is created based on RBF, so the expression of $\underline{\underline{E}}_{m,\boldsymbol{\alpha}_{\mathrm{tr}}}^{(t)}$ and $\underline{\underline{G}}_{m,\boldsymbol{\mu}_{\mathrm{tr}}}^{(t)}$ can be obtained analytically after offline training.

The reservoir is partitioned into non-overlapped subdomains, and local RBs are obtained for each subdomain via POD (as indicated in Sect. 3.3.2). The results of global POD and local POD are compared. It can be concluded that, compared to global modes, fewer local modes are needed in each subdomain for the same level of reconstruction accuracy. Note that the local ROM results are discontinuous along the subdomain interface. Thus, the fields are smoothed by solving an additional minimization problem.

Furthermore, Xiao et al. aim to solve an inverse problem. They supposed that the governing equations and solutions in a few locations of the computational domain are known. Still, the parameters that exist in the PDEs over the domain are unknown. Therefore, they aimed to compute parameter fields exploiting known solutions. This computation can be achieved via an optimization formulation that minimizes the differences between observations and computations.

The details and formulations of the smoothing algorithm and the inverse problem are beyond the focus of this review, and they are well explained in [337].

5.3.3 Hybrid Technique

A hybrid interpolation technique for inverse problems with non-overlapped decomposition is proposed by Arcucci et al. [57, 339]. The authors simplify Eq. 35 and interpolate POD coefficients as

$$\boldsymbol{\alpha}_m^{(t)} = \underline{\mathcal{Z}}_m \left(\boldsymbol{\alpha}_m^{(t-1)}, \boldsymbol{\mu}_m \right).$$

Then, the authors proposed a so-called *DD reduced order data assimilation* process to solve the inverse problem. The method results in an unconstrained least squares formulation

$$u_m = \arg\min_{u_m} \left\{ \| u_m - u_m^{(t)} \| + \| d_m^{(t)} - \underline{M} \left(u_m^{(t)} \right) \| \right\},$$

where $u_m^{(t)}$ is the ROM prediction in time (t), $d_m^{(t)}$ is the observed value of a finite number of locations in time (t), and $\underline{M}\left(u_m^{(t)}\right)$ is an observation operator that extracts values at the observed locations from $u_m^{(t)}$.

The method is applied to solve the air pollution problem of London South Bank University, modeled by 3D incompressible Navier-Stokes equations. Known measurements in several locations, $d_m^{(t)}$, are utilized to calibrate local predictions. The results reveal that the data assimilation model performs better than the standard local ROM. Note that no extra procedure is applied to smooth the local predictions, which results in discontinuities at interfaces for global fields.

5.4 Optimization-Based Technique

We turn now to the third category, the so-called optimization-based methodologies. The technique aims to formulate an optimal system to smooth the discontinuities among the subdomain interfaces.

Two methods, based on the least squares Galerkin projection and Gappy POD, are included in the group. Due to their diversity, details can be observed from the explanation below rather than a general description here.

5.4.1 Least Squares Galerkin

Choi et al. adopted the least squares Galerkin method to construct the optimal systems of domain decomposition problems [54, 172]. The numerical setups are almost the same as those presented in their previous publication [175] in Section 4.2.6. Similar objective functions are constructed, while solutions are approximated using data-driven techniques instead of Galerkin projection-based formulations.

In Eq. 22, the residuals are formulated via a Galerkin projection. In contrast, in the non-intrusive procedure shown in [54, 172], the authors use an autoencoder neural network (defined as $\underline{\mathcal{A}}$) to construct low-dimensional representations. Suppose a parameter set $\{\boldsymbol{\mu}_j\}_{j=1}^{N_\mu}$ and FOM solutions defined for interiors $u_m^{\Omega}(\boldsymbol{\mu}_j) := u(\boldsymbol{\mu}_j)\big|_{\Omega_m}$ and interfaces $u_m^{\Gamma}(\boldsymbol{\mu}_j) := u(\boldsymbol{\mu}_j)\big|_{\Gamma \in \partial\Omega_m}$. The autoencoders[26] aim to minimize the respective MSE losses

$$\mathcal{E}_m^{\Omega} = \frac{1}{N_\mu} \sum_{j=1}^{N_\mu} \left\| u_m^{\Omega}(\boldsymbol{\mu}_j) - \underline{\mathcal{A}}_m^{\Omega}(\boldsymbol{\mu}_j) \right\|_2^2, \quad \mathcal{E}_m^{\Gamma} = \frac{1}{N_\mu} \sum_{j=1}^{N_\mu} \left\| u_m^{\Gamma}(\boldsymbol{\mu}_j) - \underline{\mathcal{A}}_m^{\Gamma}(\boldsymbol{\mu}_j) \right\|_2^2$$

for internal u_m^{Ω} and surface u_m^{Γ} values of each subdomain $m = 1, \cdots, N_\Omega$.

A general formulation for approximations of any new parameter $\boldsymbol{\mu}'$ can be expressed as

$$u'^{\Omega}_m(\boldsymbol{\mu}') \approx \underline{\mathcal{A}}(\boldsymbol{\mu}')|_{\Omega_m} = \underline{\mathcal{A}}_m^{\Omega}(\boldsymbol{\mu}'), \text{ and } u'^{\Gamma}_m(\boldsymbol{\mu}') \approx \underline{\mathcal{A}}(\boldsymbol{\mu}')|_{\Gamma \in \partial\Omega_m} = \underline{\mathcal{A}}_m^{\Gamma}(\boldsymbol{\mu}').$$

Similar to Eq. 22, a ROM for a geometry $\Omega = \sum_{m=1}^{N_\Omega} \Omega_m$ is now formulated as an optimality system:

$$\min \frac{1}{2} \sum_{m=1}^{N_\Omega} \left\| \underline{\mathcal{R}}_m \left(\underline{\mathcal{A}}_m^{\Omega}(\boldsymbol{\mu}') \cup \underline{\mathcal{A}}_m^{\Gamma}(\boldsymbol{\mu}') \right) \right\|_2^2, \tag{36}$$

subject to the constraints:

$$\sum_{\Gamma \in \partial\Omega_m \cap \partial\Omega_n}^{N_\Gamma} \int_\Gamma \left| \underline{\mathcal{A}}_m^{\Gamma}(\boldsymbol{\mu}') - \underline{\mathcal{A}}_n^{\Gamma}(\boldsymbol{\mu}') \right| = 0,$$

where Ω_m and Ω_n are two adjacent partitions. This nonlinear constrained minimization problem is solved via Sequential Quadratic Programming, which is clarified in [54].

---

[26] See more details about the neural network architecture in Appendix B.5.

The method is tested on a 2D steady-state Burgers' equation with overlapping subdomains [54, 172]. The authors compare the purely data-driven approach to their earlier least-squares ROM (see Section 4.2.6). Results show that the intrusive, projection-based framework is faster, especially when combined with hyper-reduction (via mesh-element sampling), while the non-intrusive approach performs better in terms of accuracy.

5.4.2 Gappy POD Optimization

Iyengar et al. developed a different optimization framework using Gappy POD [340, 341]. This method reconstructs optimal global solutions by combining local and global POD modes. We will first outline Gappy POD [170], and then explain its implementation for approximating smooth fields in multiple subdomains.

Gappy POD is an approach to finding the optimal approximation in a domain when POD modes and values are available at a few locations. The process for achieving this is discussed as follows.

Assume a scalar field $u(\mathbf{x})$ and its POD modes $v_i(\mathbf{x})$. We have $u(\mathbf{x}) \approx \sum_{i=1}^{N_{\text{RB}}} \alpha_i v_i(\mathbf{x})$. Now, suppose that the values of u are only known at a few locations, and we aim to reconstruct the solution in the whole domain. The Gappy POD method can be used to solve this problem.

The discretized solution in the whole domain is denoted by a vector $\mathbf{u}$[27], and $\mathbf{u}^*$ represents a vector that contains the known values of $\mathbf{u}$ at a few locations and zeros elsewhere. We use a mask vector $\mathbf{m}$ to indicate the known and unknown locations, where $\mathbf{m}$ contains 1 for known values and 0 for unknown values. The goal of Gappy POD is to find a set of coefficients $\tilde{\boldsymbol{\alpha}}$ that can optimally reconstruct the solution in the whole domain using the known values. This is expressed as $\tilde{u}(\mathbf{x}) \approx \sum_{i=1}^{N_{\text{RB}}} \tilde{\alpha}_i v_i(\mathbf{x})$.

The process of Gappy POD can be regard as an optimization problem, which aims to

$$\min_{\tilde{\boldsymbol{\alpha}}} \left\| \mathbf{u}^* - \mathbf{m} \odot \left(\sum_{i=1}^{N_{\text{RB}}} \tilde{\alpha}_i \mathbf{v}_i \right) \right\|_2 .$$

The minimization problem can be solved by the least-squares method. The optimal coefficients $\tilde{\boldsymbol{\alpha}}$ are obtained by solving the following linear system:

$$\underline{\underline{M}}\tilde{\boldsymbol{\alpha}} = \mathbf{f},$$

where $M_{ij} = (\mathbf{v}_i, \mathbf{v}_j)$ and $f_i = (\mathbf{v}_i, \mathbf{u}^*)$ are the entries of $\underline{\underline{M}}$ and $\mathbf{f} = [f_1, \ldots, f_{N_{\text{RB}}}]^{\text{T}}$, respectively.

Then, the global approximation $\mathbf{u}'$ obtained via the Gappy POD, is expressed as

$$\mathbf{u}'_k = \begin{cases} \sum_{i=1}^{N_{\text{RB}}} \tilde{\alpha}_i \mathbf{v}_{i,k}, & m_k = 0 \\ \mathbf{u}^*_k, & m_k = 1 \end{cases}, \quad k = 1, \ldots, N_u,$$

where N_u is the number of discretized points in the whole domain, and $\mathbf{v}_{i,k}$ is the value of the i-th POD mode at the k-th location.

The implementation of Gappy POD for domain decomposition problems proceeds as follows. First, full-domain simulations are performed to generate global solutions. The global POD modes v_i are computed. Then, the domain is decomposed into non-overlapping subdomains, and solutions in each subdomain are extracted as local snapshots. Local POD modes are calculated for each subdomain. PODI is used to construct ROMs for each partition. As illustrated in Fig. 28, u^* is non-zero in the red areas of each division, while it is zero in the rest of the domain. Consequently, the global POD coefficients $\tilde{\alpha}$ are obtained via Gappy POD. Finally, the complete solution is reconstructed as 28. Note that the local ROMs alone produce non-smooth predictions across subdomain interfaces. Thus, the authors employ Gappy POD and global POD modes to ensure solution smoothness throughout the entire domain.

The authors validated the method using three cases: (i) quasi 1D flow through a converging-diverging Nozzle; (ii) flow over a wedge with shocks; (iii) transonic flow over an airfoil (i.e., Reynolds-Averaged Navier-Stokes equations). The comparison with the pure global ROM reveals the benefits of adopting the aforementioned local ROM technique.

5.5 Physics-Informed Neural Network

The fourth technique introduced within the data-driven category is PINNs. The procedure of PINNs to solve PDEs in

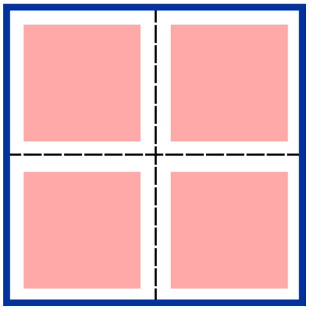


Fig. 28 Global reconstruction using Gappy POD. The entire domain is decomposed into non-overlapping subdomains (split by dashed lines). A local ROM is applied to predict results in partitions (the red region) that serve as $\mathbf{u}^*$. The rest region (the white region) is masked as unknown and approximated using Gappy POD

---

[27] We use $u(\mathbf{x})$ to denote the general (scalar) field, and $\mathbf{u}(\mathbf{x})$ for a vector field. But usually, $\mathbf{x}$ is omitted for simplicity. Here, $\mathbf{u}$ denotes a vector that contains all discretized values in the whole domain.

a single domain is already explained in Sect. 5.1.3. Note that the equation and boundary/initial condition losses also apply to domain decomposition problems. Additionally, the existence of interfaces among subdomains requires extra treatment at these interfaces to enforce the continuity of solutions over the global domain.

Since the partitions can be either overlapping or non-overlapping, this also results in different strategies for dealing with the interfaces. Thus, we categorize the two groups based on the available references. If subdomains do not overlap, the interface continuity conditions denote the equality of the variable and its flux on both sides of the face [183]. If subdomains overlap, the Schwarz iteration with the exchange of Dirichlet conditions can be employed [52]. The two scenarios are presented separately as follows.

5.5.1 Interface Constraints

For two adjoining domains with a single interface, to ensure the assembly of local approximations equal to the global approximation, both the variables and their flux should be equal on both sides of an interface [183]. Consequently, within the PINN framework, two losses are defined to satisfy the equality. Assume a subdomain Ω_m and its interface set $\mathcal{I}_m = \{\Gamma_r | \Gamma_r \in \partial\Omega_m\}$, the two terms are written by

$$\mathcal{E}_{\Gamma_m,\text{flux}} = \sum_{\Gamma_r \in \mathcal{I}_m} \frac{1}{N_{\Gamma_r}} \|q(u)|_{\Gamma_r^+} - q(u)|_{\Gamma_r^-}\|_2^2, \text{ and } \mathcal{E}_{\Gamma_m,u} = \sum_{\Gamma_r \in \mathcal{I}_m} \frac{1}{N_{\Gamma_r}} \|u|_{\Gamma_r^+} - \{u\}|_{\Gamma_r}\|_2^2,$$

where $q(u)$ denotes flux, the two sides of a interface are noted with $^+$ and $^-$, N_{Γ_r} are training points assigned at Γ_r, and $\{u\} = \frac{1}{2}\left(u|_{\Gamma_r^+} + u|_{\Gamma_r^-}\right)$ is the average operator. Thus, the local loss for Ω_m can be formulated as

$$\begin{aligned} \mathcal{E}_m = {} & w_{f,m}\mathcal{E}_{f,m} + w_b\mathcal{E}_b \\ & + w_{\Gamma_m,\text{flux}}\mathcal{E}_{\Gamma_m,\text{flux}} \\ & + w_{\Gamma_m,u}\mathcal{E}_{\Gamma_m,u}, \end{aligned} \tag{37}$$

where all terms are weighted by different factors and $\mathcal{E}_b$ appears only when Ω_m connected to global boundaries.

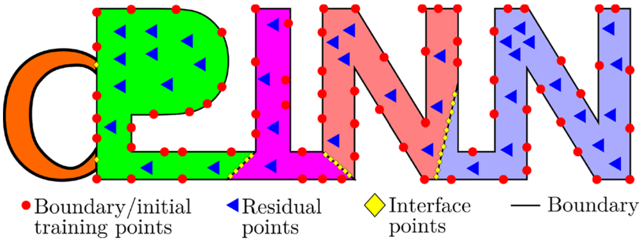


Fig. 29 Computational domain and cPINN training data points. Figure redrawn based on [318]

Be aware that a common feature of the following studies is that interface constraints are included as additional loss terms. However, the strategy of including the constraints is not unique. Over the past five years, various PINN-based architectures have been proposed. One can construct a set of local PINNs for each subdomain or a single PINN whose loss function is the sum of all partition-level losses. Moreover, FOM solutions and POD can be incorporated to reduce the complexity of the PINN structure. These aspects are described below.

5.5.1.1 Network for Physical Fields A *conservative PINN* (cPINN) is proposed by Jagtap et al. [318] to handle non-overlapping domain decomposition problems. The model uses local neural networks to approximate each subdomain, yielding local error formulation for Ω_m as $\mathcal{E}_m = w_{f,m}\mathcal{E}_{f,m} + w_{b,m}\mathcal{E}_{b,m} + w_{\Gamma_m}\left(\mathcal{E}_{\Gamma_m,\text{flux}} + \mathcal{E}_{\Gamma_m,u}\right)$.

The cPINN was used to solve both forward and inverse problems of various nonlinear PDEs, including Burgers', Korteweg-De Vries, incompressible Navier-Stokes, and compressible Euler equations. The computational domain and the location of training points are displayed in Fig. 29. The whole geometry is decomposed into several subregions, and the two additional errors are computed at the interface points.

The authors tested and compared the effect of PINN structures, including the number of hidden layers, the number of neurons per layer, the activation functions, and the size of training points. Their numerical experiments for the 1D Burgers' equation demonstrate that increasing the number of hidden layers and neurons can improve accuracy. A key advantage of a cPINN is that different network architectures can be defined for each subdomain to better capture local physical phenomena, thereby balancing the training cost and accuracy.

Jagtap et al. further developed the cPINN and proposed the *eXtended PINN* (XPINN), which can be extended to any PDEs (both forward and inverse problems) with non-overlapping partitions [342, 343]. The key modification of XPINNs is that a residual continuity loss is added at interfaces. For example, given a set of interfaces $\mathcal{I}_m$ for a subdomain Ω_m, the total residual continuity loss is defined as

$$\mathcal{E}_{\Gamma_m,f} = \sum_{\Gamma_r \in \mathcal{I}_m} \frac{1}{N_{\Gamma_r}} \left\| \underline{\mathcal{D}}_{\Omega_m}\left(u'(\mathbf{x},t)\right)\Big|_{\Gamma_r^+} - \underline{\mathcal{D}}_{\Omega_n}\left(u'(\mathbf{x},t)\right)\Big|_{\Gamma_r^-} \right\|_2^2,$$

where $\mathcal{I}_m$ is the set of interfaces for subdomain Ω_m, $\underline{\mathcal{D}}_{\Omega_m}$ and $\underline{\mathcal{D}}_{\Omega_n}$ are the governing PDEs defined in Ω_m and Ω_n, respectively, $\Gamma_r = \partial\Omega_m \cap \partial\Omega_n$, N_{Γ_r} training points is assigned at Γ_r, $+$ and $-$ denote the two sides of Γ_r.

The combined local MSE is now denoted by $\mathcal{E}_m = w_{f,m}\mathcal{E}_{f,m} + w_{b,m}\mathcal{E}_{b,m} + w_{\Gamma_m}\left(\mathcal{E}_{\Gamma_m,f} + \mathcal{E}_{\Gamma_m,u}\right)$. Note that the equality and residual continuity terms can sufficiently enforce the XPINN approximation to satisfy the global governing PDE. Thus, the flux matching constraints $\mathcal{E}_{\Gamma_m,\text{flux}}$ are not necessary and can be optionally imposed depending on the PDEs.

The cPINN and XPINN were compared in terms of parallel efficiency for 1D Burgers' and 2D steady-state incompressible Navier-Stokes equations [344]. The results show that cPINNs are more efficient for spatial decomposition problems. In contrast, XPINNs are more flexible for temporal domain decomposition and are additionally well-suited for arbitrarily shaped and complex subdomains.

Dwivedi et al. proposed a different framework for solving forward and inverse problems of several PDEs among non-overlapping subdomains [322], which is called a *Distributed PINN* (DPINN). Their method integrates temporal decomposition and trains separate networks for each time step. This approach requires an additional temporal continuity loss $\mathcal{E}_\text{t}$ to match two sequential time steps $(t-1)$ and (t) , which is given by

$$\mathcal{E}_\text{t} = \frac{1}{N_t}\|u^{(t)} - u^{(t-1)}\|_2^2.$$

Additionally, the total loss function of a DPINN is distinct from previous architectures. The authors also define the individual network for each subdomain. However, the DPINN doesn't define $\mathcal{E}_m$ for each subdomain. Instead, the total loss is expressed as a summation of all $\mathcal{E}_m$, namely:

$$\mathcal{E} = \sum_{m=1}^{N_\Omega} w_{f,m}\mathcal{E}_{f,m} + w_b\mathcal{E}_b + \sum_{r=1}^{N_\Gamma}\left(w_{\Gamma_r,u}\mathcal{E}_{\Gamma_r,u} + w_{\Gamma_r,\text{flux}}\mathcal{E}_{\Gamma_r,\text{flux}}\right) + \sum_{t=1}^{N_t} w_t\mathcal{E}_t,$$

where N_Ω, N_Γ, and N_t are the number of subdomains, interfaces, and time steps, respectively.

The DPINN is tested in several 2D benchmark problems, including steady heat conduction, Laplacian, Burgers', advection, and steady Navier-Stokes equations.

The PINNs discussed above utilize the strong form for loss formulation. Given the prevalence of weak formulations based on Galerkin methods in the intrusive framework, Kharazmi et al. developed the *hp*-variational PINN (*hp*-VPINN) [323] for nonoverlapping divisions. In this approach, the trial space is represented globally by a single neural network across the computational domain. The piecewise polynomials (i.e., Legendre polynomials, as mentioned in the work) are applied separately as test functions for every partition. In this setting, the VPINN achieves h-refinement through increased spatial resolution and p-refinement via higher-order polynomial test functions. One can analogize this framework to FEM employing the Petrov-Galerkin projection. Each subdomain can be regarded as an element of FEM, and different test and trial spaces are adopted.

Instead of multiple local PINNs, a single PINN is built and trained to approximate the global solution. In such a case, the total loss function of the *hp*-PINN, $\mathcal{E}^{hp}$, is a summation of all local losses $\mathcal{E}_m$ as shown in Eq. 37. Thus,

$$\mathcal{E}^{hp} = (\mathcal{E}, w), \text{and } \mathcal{E} = \sum_{m=1}^{N_\Omega} w_{f,m}\mathcal{E}_{f,m} + w_b\mathcal{E}_b,$$

where $(\cdot,\cdot)$ denotes the inner product operation, w is the test function, and N_Ω is the number of subdomains. Note that only a single PINN is employed to approximate the entire solution, and the polynomials test functions are continuous at interfaces. Thus, no interface constraints are required in $\mathcal{E}^{hp}$.

An example of local test functions and the solution is illustrated in Fig. 30. The test functions are only defined inside the corresponding subdomains. All local solutions are *glued* together to form a global result.

The efficiency and accuracy of the *hp*-VPINN were validated and compared with a PINN in Poisson's equation and the advection-diffusion equation. The results of 2D Poisson's

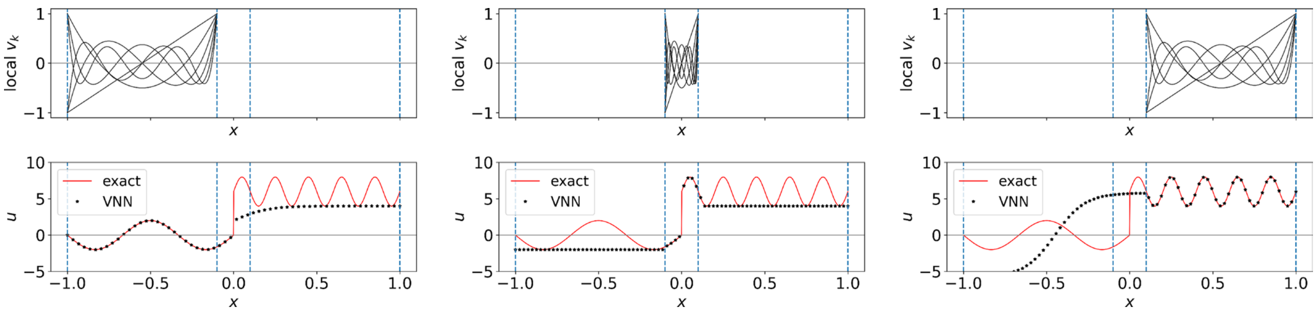


Fig. 30 Local test functions of a VPINN over each subdomain (first row). Results of each subdomain using the VPINN (second row). The dashed blue lines are the sub-domain boundaries. Figures redrawn based on [323]

equation demonstrate that the *hp*-VPINN with domain decomposition performs better than the standard PINN, and the *h*-refinement (increasing the number of subdomains in specific regions) yields a better local approximation.

5.5.1.2 Network Employing POD Modes Rather than directly approximating solutions, Xinyu Pan and Dunhui Xiao proposed a novel approach that employs neural networks to obtain POD coefficients, called the *Domain Decomposition Physics-Data Combined Neural Network* (DD-PDCNN) [345]. The DD-PDCNN loss function is similar to the cPINN, that is

$$\mathcal{E} = w_f\mathcal{E}_f + w_b\mathcal{E}_b + w_I\mathcal{E}_I + w_{\Gamma,\text{flux}}\mathcal{E}_{\Gamma,\text{flux}} + w_{\Gamma,u}\mathcal{E}_{\Gamma,u},$$

where $\mathcal{E}_I$ and $\mathcal{E}_b$ represents the initial and boundary condition terms, respectively. These terms are similar to those of a cPINN, but they are formulated in terms of POD modes rather than solutions.

Note that the usage of POD modes $\mathbf{u} = \sum_{i=1}^{N_{\text{RB}}} \alpha_i \mathbf{v}_i$ yields substantial computational savings in computing loss functions for the PINN. For example, assume a residual $\mathcal{E}_f = \Delta\mathbf{u}$, the error can be rewritten as $\Delta\mathbf{u} = \sum_{i=1}^{N_{\text{RB}}} \alpha_i \Delta\mathbf{v}_i$. Hence, $\Delta\mathbf{v}_i$ can be pre-computed and fixed for the training. The losses in each training step can be obtained by multiplying them by the POD coefficients.

The technique was applied to three cases: the Korteweg-de Vries equation, Kovasznay flow, and the incompressible Navier-Stokes equations. In comparison to the global PDCNN, the DD-PDCNN comprised fewer hidden layers and neurons, and it is far more accurate. This finding is consistent with the aforementioned PINN-based architectures. A disadvantage of this approach is that it requires high-fidelity solutions to compute POD modes.

5.5.2 Schwarz Iteration

The last sub-category of the data-driven techniques combines the PINN and the Schwarz method. According to the literature, the combination is not unique. It can be applied to the training stage, in which the values at overlapping regions are exchanged between adjacent domains. Also, one can construct PINN-based ROMs and then couple them iteratively via the Schwarz algorithm. The details of the two procedures are presented as follows.

5.5.2.1 Schwarz for Training PINNs The Schwarz procedure is employed here to iteratively assign local boundary values of multiple subdomain-level PINNs. Consequently, for Ω_m, the loss function is given by

$$\mathcal{E}_m = w_{f,m}\mathcal{E}_{f,m} + w_{b,m}\mathcal{E}_{b,m},$$

where $\mathcal{E}_{b,m}$ is error of local boundaries. Suppose $\Gamma_{[mn]} = \Omega_n \cap \partial\Omega_m$ and the additive Schwarz method is employed, $\mathcal{E}_{b,m}$ is computed by

$$\mathcal{E}_{b,m} = \frac{1}{N_{\Gamma_{[mn]}}} \left\| \mathbf{u}_m^{(t)}\Big|_{\Gamma_{[nm]}} - \mathbf{u}_n^{(t-1)}\Big|_{\Gamma_{[nm]}} \right\|,$$

where (t) denotes the step of training, and $N_{\Gamma_{[nm]}}$ is the number of training points at $\Gamma_{[nm]}$.

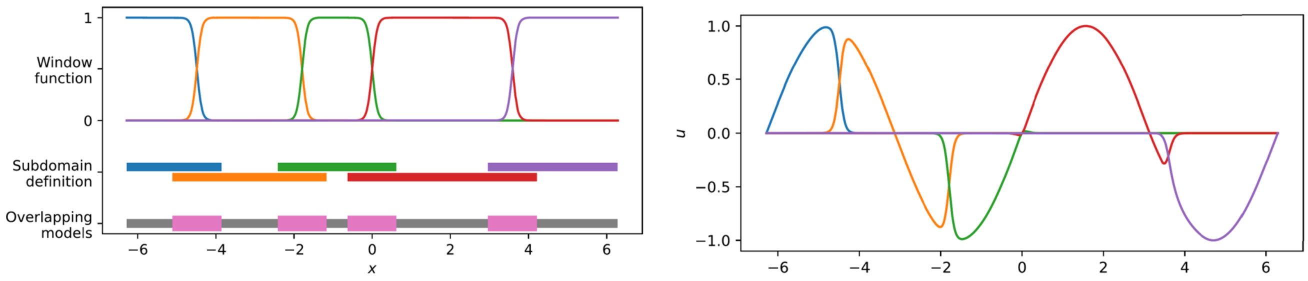
Li et al. [346] applied this procedure to model two elliptic equations - a classical Poisson equation and a steady-state Schrödinger equation - considering an overlapping decomposition. They initiated separate PINNs for each subdomain. Then, in each iteration of the training, values of overlapping regions are shared with neighbours.

This approach is known as the *deep domain decomposition method*. It is worth noting that, in their study, the PDEs loss $\mathcal{E}_f$ is formulated concerning an optimization system of the problem, instead of the residual presented in Sect. 5.1.3.

A nearly identical framework to the deep DD method was developed by Li et al. for elliptic problems [347] with overlaps. They investigated the influence of several factors on the performance of the solver. The accuracy of different PINN structures is compared, including the number of hidden layers, the number of neurons per layer, and the size of the training points. Different sizes of overlap and partitions are also compared. Within the range of their tests, more complex neural networks and data points, as well as more subdomains, yield more accurate ROM approximations. The accuracy increases with the overlapping region, which is already proven for intrusive methods [53].

Moseley et al. developed a so-called *Finite Basis PINN* (FBPINN) that enables higher flexibility for the training stage. [324]. The workflow of a FBPINN can be summarized into three steps. Firstly, multiple small neural networks are instantiated for each subdomain. Each local solution is weighted by a so-called window function, which is smooth, differentiable, and vanishes outside the specific subregion. Since partitions overlap, as well as window functions, the solutions of neighbouring parts are incorporated into the MSE of each subproblem (i.e., Schwarz method). Then, all local PINNs are trained simultaneously. Finally, in the online stage, their predictions are weighted by the window functions to assemble a global solution.

An 1D example of the FBPINN solver is shown in Fig. 31. The entire model is decomposed into overlapping



(a)

(b)

Fig. 31 One dimensional example of the FBPINN solver. (**a**) Domain decomposition and window functions for each subdomain. (**b**) Local solutions. Figures taken from [324], copyright owned by Springer

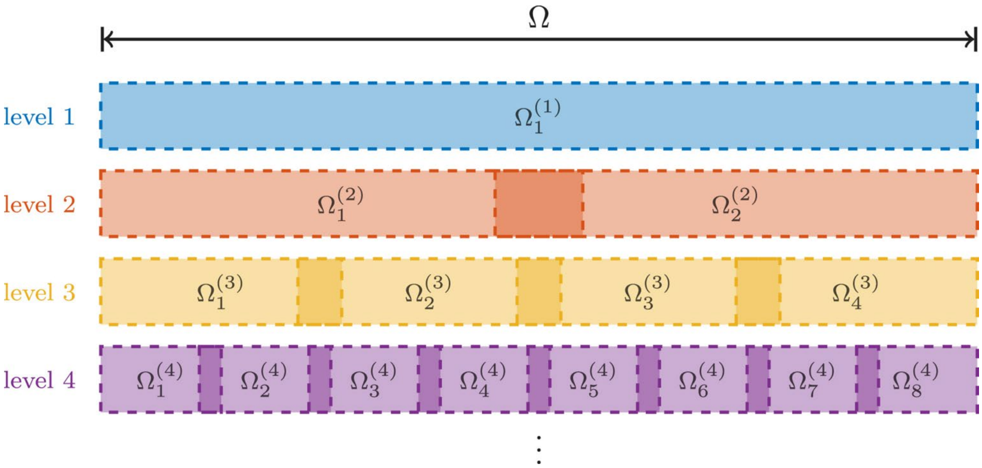


Fig. 32 Hierarchy of the four-level domain decomposition used in the multilevel FBPINN. Figure taken from [350], copyright owned by Elsevier

subdomains. Separated window functions are defined for each partition. Local results are weighted by corresponding window functions and summed to form total solutions.

The authors also present a *flexible* training schedule, where only a subset of local FBPINNs is trained at each iteration, rather than all of them. This strategy allocates computational resources by focusing on complex locations while reducing efforts for simpler regions, thereby achieving an effective balance between accuracy and computational efficiency.

The FBPINN is applied to approximate the first- and second-order differential functions, and then utilized to model Burgers' and wave equations. A comparison with the standard PINN (see Sect. 5.1.3) reveals that the FBPINN converges faster and results in lower final error. Additionally, each local FBPINN has fewer layers and neurons compared to the global PINN, which can significantly reduce training cost.

A *multilevel FBPINN* (MFBPINN) was proposed by Dolean et al. [348–350], combining the FBPINN framework [324] with the Schwarz iteration. They weighted all local losses using window functions and united them as a global PINN loss. At each iteration, only a subset of subdomains is activated and trained. Then, the updated values on the overlapping regions are shared with neighbouring subdomains, as in the Schwarz method. The novelty of this approach is that more computational resources are allocated to train portions with higher losses. The strategy achieves a balance between cost and convergence, thereby improving overall computational efficiency.

An innovative aspect of this method compared to previous PINNs is that the geometry is divided into different numbers of subdomains. The authors utilized four decomposition *levels*, as shown in Fig. 32.

The MSE of partitions at all levels is weighted by window functions and then summed as the global loss function. The authors indicated that employing the configuration can improve the numerical scalability and convergence properties.

The algorithm is applied for the simulation of several problems: 1D ordinary differential equations, Laplacian problems, and the Helmholtz Equation. The results demonstrate that very small overlapping sizes affect the convergence. A multilevel FBPINN with fewer hidden layers and neurons can achieve an error several orders of magnitude lower than the standard PINN.

For the PINN architectures mentioned above, both local and global boundary conditions are constrained via loss functions. Snyder et al. published an investigation that focuses on parameterizing the Dirichlet BCs [351].

In this approach, the solution on a subdomain Ω_m is represented as

$$u'_m(\mathbf{x};\boldsymbol{\mu}) = \gamma_m(\mathbf{x})\boldsymbol{\theta}_m(\mathbf{x};\boldsymbol{\mu}) + \varrho_m(\mathbf{x})g_{D,m}(\mathbf{x}),$$

where $\boldsymbol{\theta}_m(\mathbf{x};\boldsymbol{\mu})$ is the neural network approximating the solution over Ω_m, $\gamma_m(\mathbf{x})$ is a smooth function (can be referred as a test function) that vanishes on $\partial\Omega_m$, $g_{D,m}(\mathbf{x})$ represents the solution at the overlapping regions transferred from neighboring subdomains and $\varrho_m(\mathbf{x})$ serves as the weighting function defined at the overlaps. The authors indicated that this formulation can *strongly enforce* Dirichlet BCs. In previous PINNs, Dirichlet BCs are added as penalty terms in the loss function, which is regarded as a *weak enforcement*.

The authors compared three configurations: (i) weak imposition of both global BCs and interfaces; (ii) strongly constrained conditions for all Dirichlet boundaries; (iii) mixed approach combining strong enforcement of global BCs and weak imposition at interfaces. The results of a 1D advection-diffusion problem reveal that the mixed scheme (iii) converged faster and provided a more accurate solution compared to the other two strategies. Moreover, they successfully coupled a PINN-ROM and a FOM within the framework.

5.5.2.2 Schwarz for Coupling PINN ROMs We remind you that the PINN with Schwarz architectures reviewed above can only predict solutions for the specific domain in which they are trained. To address this shortcoming, Feeney et al. [352] developed a framework that utilizes the Schwarz algorithm to couple multiple local PINNs in the online stage.

The approach is implemented as follows. In the offline stage, a PINN is constructed to solve *boundary value problems* in a small subdomain with arbitrary Dirichlet BCs. Then, in the online stage, local ROMs are iteratively assembled into a global approximation using the Schwarz method. We remark that the procedure is similar to the *boundary-to-boundary mapping* presented in Sect. 5.2.

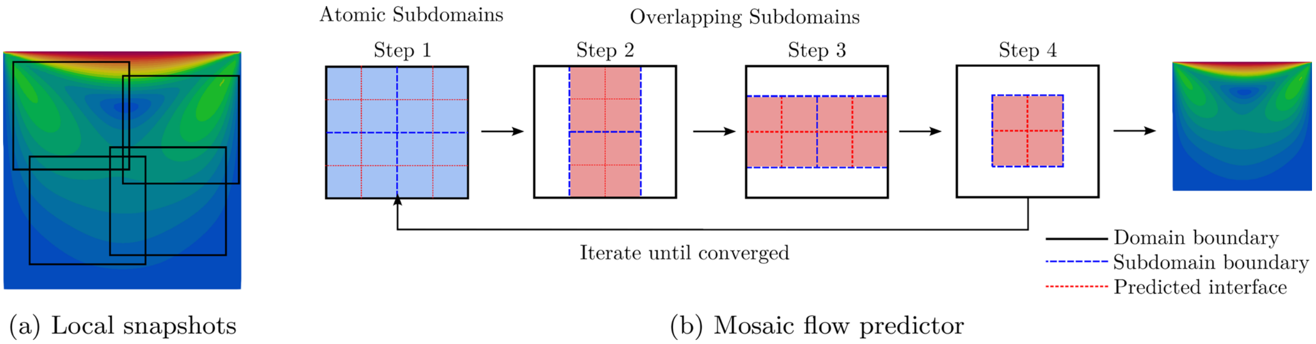
The authors proposed two tools to achieve the two steps: (i) a PINN-based *subdomain solver network* to solve boundary value problems; (ii) a so-called *mosaic flow predictor* to couple local solvers. The combined architecture is referred to as the *distributed mosaic flow predictor*, and the two stages are illustrated in Fig. 33. The oversampling strategy (see Sect. 3.3.3) is applied to generate parametric high-fidelity solutions. Then, various local snapshots are collected and used for training. To achieve BC parameterization, the input layer of the subdomain solver network includes not only the spatial and temporal variables ($\mathbf{x}$ and t) but also BC parameters.

The *mosaic flow predictor* is referred to as the iterative scheme shown in Fig. 33b. Each iteration contains four sub-steps. The atomic subproblems are solved to obtain a global approximation. Then, the solutions of the so-called *overlapping subdomains* are computed to accelerate the propagation of global boundary information into the interior regions, i.e., speed up the iterative process. The two types of subdomains overlap with their neighbours, and their local BCs are exchanged following the Schwarz iteration.

The authors apply the method to analyze a 2D Laplace equation. Efforts are devoted to testing the accuracy and parallelization performance using different graphics cards. They have proven that it is capable of predicting a very large domain comprising up to 32 × 32 partitions.

It is worth noting that, in [353], a *genomic flow network* based on deep neural networks was created as an alternative to replace *subdomain solver network*. The coupling of ROMs was also achieved via *Mosaic Flow Predictor*.

The numerical experiments indicate that the ROM trained on a small piece can be applied to predict significantly larger domains. For Laplace and Navier-Stokes equations, the size of the total domain is 1200 times and 12 times that of a subdomain, respectively. The training of the *genomic flow network* is 1–3 orders of magnitude faster compared to the



(a) Local snapshots (b) Mosaic flow predictor

Fig. 33 A sketch of the *distributed mosaic flow predictor*. (**a**) FOM simulations are performed in a slightly larger model, and local snapshots are extracted from the solutions, which is similar to the oversampling technique explained in Sect. 3.3.3. (**b**) *mosaic flow predictor* contains four sub-steps in each iteration. Both atomic and overlapping subdomains overlap with their neighbours, and the solutions are communicated with the Schwarz iteration. Figures redrawn based on [353] standard PINN. Moreover, its accuracy is significantly better compared to the XPINN.

We highlight that applying the Schwarz algorithm for coupling many subdomains can reduce the cost of training PINNs. In contrast, using the Schwarz iteration for a PINN training does not avoid the computational expense over the global domain.

Lastly, we would like to note that PINNs have also become popular for solving multiphysics problems. However, in the papers we noticed, monolithic networks are constructed to predict flow and structural dynamics, which can not be completely considered as local ROMs. As a hint, we list some references for different applications: (i) FSI problems [354, 355]; (ii) multiphysics in chemical engineering, including Navier-Stokes, energy conservation, mass transport and chemical kinematic equations [356, 357]; (iii) electro-thermal coupling (two Laplace equations) [358].

6 Summary and Conclusions

In this paper, we have tried to present an overview of the *local ROM methodology*: a challenging and rapidly developing field. We aim to provide an informative and brief description, avoiding the complexities of a strict mathematical derivation that can be found anyway in the original literature. This approach, which may be too shallow for some readers, has helped us create a narrative that makes this paper possible, considering the broad scope of the methods.

Considering the sources consulted, we realize that *domain decomposition* in frames of *model order reduction* techniques, is dominated by a fundamental ideology: minimize the jumps across the subdomain interfaces, meanwhile satisfying the governing PDEs and the boundary conditions.

From our literature review, we may conclude that the available techniques are based on a few concepts. However, such a general overview may be shadowed by an extensive amount of individually readapted techniques that are proposed and consequently modified to fulfill the authors' necessities. In summary, we may deduce that the available methods can be adequately classified into two categories: (i) *intrusive*; (ii) *non-intrusive*. The former involves manipulating the original PDEs to derive reduced-dimensional systems. In contrast, the second group is constructed purely based on FOM solutions.

We note that the strategy for domain decomposition and the computation of local RBs are prerequisites of any technique. Thus, an overview of them is presented first. Then, considering the distinct principles between the two groups of coupling techniques, we will summarize their research status separately.

6.1 Common Aspects

We may summarize that the selection of *overlapping* or *non-overlapping* of subdomains and the usage of *conforming* or *non-conforming* mesh is a choice incorporated in the specific coupling methodology. The choice depends on the following three criteria: (i) overlapping decomposition has better numerical characteristics; (ii) non-overlapping subdomains appear to be more flexible for more complex and realistic problems. It also reduces the total amount of interfaces; (iii) non-conforming meshes are suitable for problems in which spatial resolution differences are more pronounced.

Regarding the procedure utilized for the decomposition, we may formulate a couple of conclusions. *Individual decomposition* is a very flexible technique, appealing to a wide range of scenarios. In spite of this, we believe that the other category, the so-called *generic decomposition*, will be more attractive to realistic applications in the next years. This is due to its capabilities to represent geometries with repeating divisions.

Before performing a series of high-fidelity simulations, the parameters should be determined. *Sampling strategies* are critical for minimizing the number of simulations while maximizing the information obtained from them. For this purpose, different algorithms have been proposed, depending on the dimension of the parameter space. Both physical and geometrical features can be considered as parameters. The former is more straightforward, while the latter is more complex and challenging. Thus, *geometrical parameterization* strategies are discussed. Additionally, the parameter space can be very large when considering too many factors, which complicates the optimization process. Hence, *parameter space reduction* techniques are necessary to reduce the dimension of the parameter space. Another parameterization issue is the wide parameter ranges, which can lead to a large variance in the solution manifold and consequently require too many basis vectors to capture the solution space. In this case, *local RBs* in the parameter/solution space can be computed with respect to several clusters of parameters, which can improve the overall accuracy of the RBs.

After parameterization, it turns to the stage of *snapshots collection* and *local RB construction*. We may conclude from the engineering point of view that: (i) *localized global RB* may not be attractive due to their expense in the offline stage; (ii) *global solutions and local RB* may be applied for some specific problems due to the ability of local basis vectors to capture small-scale phenomena. Nevertheless, it will show limited growth due to the extensive cost of training. (iii) *Localized training and oversampling technique* is a very promising technique for large-scale facilities due to its inexpensive offline stage cost and capabilities to represent subdomain-level variance via the local RBs.

Lastly, to further accelerate ROMs, *incremental* strategies and *hyper-reduction* techniques have been proposed. Both approaches apply to intrusive and non-intrusive methods, and to global and local ROMs. For incremental ROMs, the crucial factors are the choice of the initial basis and the basis enrichment strategy. For hyper-reduction, the *discrete empirical interpolation method* is among the most widely used techniques.

6.2 Coupling Techniques

6.2.1 Projection-Based Methods

Now, we turn our attention to the intrusive coupling approaches. They denote reduced algebraic systems explicitly derived from the original PDEs, and then solved numerically. The available intrusive methods can be categorized into two sub-groups (see Table 2): (i) *monolithic*; (ii) *iterative*.

In turn, these two are characterized by a few conceptions. Monolithic techniques are based on one of the following three ideas: (i) use specific interface basis – test and/or trial – to enforce continuity; (ii) create a scheme to minimize by penalty or constrain the discontinuity of local basis vectors; (iii) construct a continuous global basis using some strategy to impose a coherent superposition. Even simpler, *iterative* algorithms follow a single conception: create a set of local problems, resolve them iteratively by exchanging information through interfaces, and continue until jump and error thresholds are satisfied.

From our literature review, we have observed that techniques belonging to the monolithic group can be arranged into ten categories, according to their ideology. Iterative techniques can be further classified according to the type of interface condition. Table 2 summarizes the properties of each technique according to the classification addressed in Sect. 2. Readers can easily notice the variety of problems to which each of the methods has been applied. Most of the techniques have been utilized for elliptic and parabolic equations. This is probably because these basic problems are used to test and validate the methods.

Considering the monolithic techniques, we highlight that the method that inherits the concept of RBEM (i.e., RBEM, RDF, SCRBEM, DGRBEM) is more prospective for application. They share two significant advantages: (i) A localized training strategy is incorporated to reduce the offline cost; (ii) Entire models are assembled by instantiations transformed from archetype subdomains, and geometrical parameterization can be exploited during the transformation.

Moreover, we believe *DGRBEM* is the most promising for engineering problems. This is due to its advantages in the following four aspects: (i) It utilizes localized training, resulting in an efficient offline stage. (ii) Although DG is quite novel in the field of assembling local ROMs, we consider that it is mature because it has been developed for many years as a high-order method. (iii) It incorporates a more compact RB, including interior and domain interfaces. The discontinuities of the basis vectors are involved in the DG framework. Thus, no extra procedures are needed to obtain separated interface spaces and ensure continuity. (iv) FOM and ROM formulations are consistent, differing only in their basis functions. DG-FOMs employ polynomial bases in the discretized finite elements, whereas DGRBEM uses reduced bases in each subdomain. This facilitates coupling FOMs and ROMs in practice.

The three categories organized under the item of iterative algorithms are inherently and, in comparison, not so distinct. In spite of this, a preferred method can be decided upon based on the conditions at the boundary. Dirichlet-Dirichlet iterations generally require overlapping subdomains. This introduces additional difficulties for complex geometries. Therefore, we forecast a trend to avoid Dirichlet-Dirichlet coupling and choose Dirichlet-Neumann or Robin-Robin configurations for engineering applications. We believe that the most important challenge in the near future will be to propose efficient schemes for the communication between sub-problems for assembling numerous local FOMs and ROMs.

Regarding the comparison of monolithic and iterative techniques, we can forecast the following trends. *Monolithic* methods are, in general, more complex and intrusive. This complexity arises when deriving reduced systems with better numerical features. Unfortunately, this also results in additional difficulties in its implementation. They must be extensively redeveloped in accordance with the specific demands of the problem. In contrast, *iterative* methods enjoy a more general formulation. They are also more prone to assembling numerous high-resolution systems and ROMs, as well as multi-physics scenarios. Therefore, this may allow us to foretell that the implementation of iterative techniques may occur sooner in commercial codes and industrial applications. In contrast, *monolithic* may remain a long-term path for the simulation of engineering problems. A common trend of both approaches is that they should be further developed to be suitable for a large amount of subdomains.

6.2.2 Non-Intrusive Methods

Recently, pure data-driven methodologies have been increasingly utilized to construct ROMs, as well as for local ROMs. Based on our observations, we can conclude that the principle of non-intrusive ROMs aims to establish parameter-solution mappings. This framework considers

Table 2 Intrusive local ROMs. Classifications with respect to domain decomposition, parameterization, RB construction and applications. Abbreviations: *Arche.* (archetype), *CV-Diff.* (convection-diffusion), *DG* (discontinuous Galerkin), *Diff.* (different), *DD* (domain decomposition), *D-D* (Dirichlet-Dirichlet), *D-N* (Dirichlet-Neumann), *FOM-ROM* (a FOM couple with a ROM), *GFEM* (generalized FEM), *LSG* (least squares Galerkin), *MsFEM* (multiscale FEM), *N-N* (Neumann-Neumann), *Optim.* (Optimization-based), *PR* (port reduction), *PUM* (Partition of Unity method), *RB* (reduced basis), *RBEM* (reduced basis element method), *R-R* (Robin-Robin), *ROM-ROM* (a ROM couple with a ROM), and *SC* (static condensation)

			Monolithic											Iterative			
			RBEM	RDF	SCRBEM		DG-based local ROM		GFEM/PUM		Optimization		Multi- physics	D-D	D-N	R-R	Multi- physics
					PR	SC	DGRBEM	Local DG	PUM	MsFEM	Optim.	LSG					●
DD	Adjacent	Overlapping							●		●	●	●	●			●
		Non-overlapping	●	●	●	●	●	●		●			●	●	●	●	●
	Mesh	Conforming	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
		Non-conforming	●				●						●	●	●		●
	Subdomain	Individual						●			●	●	●	●	●	●	●
		Archetypes	●	●	●	●	●		●	●	●	●				●	
Parameters		Physical	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
		Geometrical	●	●	●	●	●	●			●		●	●			●
Snapshots & RB	Global	Localized RB						●									
		Diff. local RB						●			●	●	●	●	●	●	●
		Arche. local RB									●						
	Local	Arche. RB	●	●	●	●	●		●	●						●	
Equations	Elliptic	Second order	●	●		●	●	●	●	●	●	●	●	●	●	●	●
		Helmholtz	●			●											
	Parabolic	Second-order		●				●	●	●	●		●		●	●	●
		CV-Diff.				●	●							●	●	●	
	Hyperbolic	Wave												●			
		Transport								●							
	Other	Maxwell's	●				●										
	Fluid	Burgers'										●		●			
		Euler						●					●	●			
		Stokes	●	●			●			●						●	
		Navier stokes	●		●		●		●	●	●		●	●			●
Coupling		ROM-ROM	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
		FOM-ROM						●						●	●		

two essential aspects: (i) the algorithms employed for constructing surrogate models; (ii) input and output data. Apart from the parameters (e.g., physical properties and boundary conditions), the interface values between subdomains are also involved accordingly as inputs and/or outputs of the local ROMs.

Considering the procedure of coupling, the non-intrusive local ROMs can be categorized into four groups. A summary regarding their applications is listed in Table 3.

There are two subcategories that incorporate interpolations to achieve global approximations. They follow a similar procedure. Independent ROMs are constructed for each partition, and their fields are updated iteratively with inputs from adjacent subdomains. The difference is the quantities exchanged between subproblems. For Schwarz-based approaches, interface values from neighbours are considered as inputs. In contrast, for interpolation-based methods, the solutions of adjacent subdomains are adopted to update each local ROM.

The optimization-based techniques seem more *monolithic*. They intend to minimize interface discontinuities among all subdomains. The optimal systems are solved using data-driven approaches.

PINN-based ROMs demonstrate greater diversity. Those imposing extra interface constraints can be regarded as *monolithic*, in which networks are built to diminish residuals and interface jumps simultaneously. Alternatively, the Schwarz iteration can be incorporated to train PINNs, computing local residuals along the interfaces. Additionally, PINNs can be used to create boundary value problem solvers (local ROMs) and then couple subdomain-level problems.

Regarding the techniques, we believe the Schwarz-based and iteration approaches are more practical for industrial practices. Our opinion is based on the following three points: (i) They are very flexible in employing different dimensionality reduction algorithms and surrogate models; (ii) They are less expensive than PINNs (still time-consuming nowadays) in the offline stage; (iii) According to the observed studies, they support both overlapping and non-overlapping decompositions.

Now, let us discuss the PINN framework. It is still not practical for large-scale applications due to the prohibitively high computational cost for training neural networks. However, it contains physical knowledge and demonstrates its potential for solving inverse problems. As the computational hardware is growing recently, it might become popular in the near future.

6.2.3 Complex Systems

The coupling of multiple local ROMs is also well-suited for *multiphysics* problems, where each physics can be modeled with an individual ROM. Three types of complex systems are described in the review: (i) the *fluid-structure interaction* is one of the most common scenarios; (ii) *multiphysics* phenomena comprising fluid dynamics, structural mechanics, electronics, neutronics, etc; (iii) *bifurcations* in fluid flows. Especially for FSI problems, two innovative techniques are

Table 3 Non-intrusive local ROMs. Methods, classification of domain decomposition, RB construction (or dimensionality reduction) and applications. Abbreviations: *PINN* (Physical informed neural network)

			Schwarz	Interpolation	Optimization	PINN	
						Loss term	Iterative
DD	Adjacent	Overlapping	•	•	•	•	•
		Non-overlapping	•	•	•	•	•
	Mesh	Conforming	•	•	•		
		Non-conforming	•				
	Subdomain	Individual	•	•	•	•	•
		Archetypes	•	•			•
Parameters		Physical	•	•	•	•	•
		Geometrical			•		
Snapshots & RB	Global	Diff. local RB	•	•	•	•	•
		Arche. local RB		•			
	Local	Arche. RB	•				•
Equations		Elliptic	•	•		•	•
		Parabolic	•	•		•	•
		Hyperbolic					
	Fluid	Burgers'	•		•	•	
		Euler				•	
		Stokes					
		Navier stokes	•	•	•	•	•
Coupling		ROM-ROM	•	•	•	•	•
		FOM-ROM	•		•		

included: the reduction of mesh motion using interpolation algorithms and the embedded boundary method.

Both intrusive and non-intrusive frameworks have been employed for simulating those conditions. For multiphysics processes, the variables are generally confined only to specific subregions. Thus, the aspects of generic spatial decomposition, localized training strategy, and RBs defined in reference domains do not always apply to them. Nevertheless, the same algorithms can still be utilised to assemble the ROMs, such as *Lagrange multiplier* and *iterative* schemes. Besides, in some situations, homogeneous conditions are defined at the interfaces, and then the subdomains can be coupled straightforwardly.

6.3 Conclusions

According to the literature review and aforementioned summaries, we may conclude that:

- Recent developments in local ROMs have indicated a shift from intrusive to non-intrusive methodologies. The former have been developed for around two decades. In contrast, most machine learning-based studies appeared within the last five years. However, due to their advantages and flexibility, they are becoming popular, especially those utilizing neural networks.
- Iterative algorithms are well-suited for coupling local ROMs, in both intrusive and non-intrusive frameworks. Since each subproblem can be constructed independently, it is capable of coupling different sub-models, including different ROMs, FOMs with ROMs, and multiphysics.
- Many large-scale engineering devices are assembled by numerous repeating subdomains. Thus, the generic domain decomposition and localized training procedures are very promising. They have the potential to reduce the computational cost during the offline stage and accurately reconstruct global high-fidelity solutions.
- Regarding parameterization, various strategies have been investigated for different situations. Geometrical parameterization and shape optimization have also been well studied in the past decades. RBF, IDW, and FFD are feasible tools for accomplishing this task. Recently, advanced parameter space reduction employing AI and neural networks is full of potential to improve the accuracy and efficiency of ROMs.
- The local ROM framework has been widely applied to single-physics problems for decades. It is also well-suited to complex phenomena, such as fluid-structure interaction and bifurcations in fluid flow. Additionally, two innovative techniques, e.g., embedded method and interpolation algorithms, have been proposed for the geometric reduction of FSI problems. In the near future, the trend will be toward more challenging *multiphysics* phenomena involving fluid dynamics, structural mechanics, electronics, neutronics, etc.
- Projection-based approaches contain more comprehensive mathematical foundations, e.g., a priori and a posteriori error estimations, convergence, and well-posedness analyses. They are also more mature in applications, such as geometrical parameterization, FOMs-ROMs coupling, and multiphysics problems.
- For non-intrusive methods, the procedure is similar for different scenarios, thus they can be easily adapted to solve various problems. Unlike intrusive techniques, which require manipulation of governing equations, they have to be redeveloped with respect to PDEs and boundary conditions.

Lastly, we consider it may be adequate to mention that the aforementioned opinions and prognoses are mainly based on an engineering point of view. This might tend to underestimate very important mathematical aspects, such as well-posedness and error estimations. We hope for the indulgence of those readers regarding our conclusions as controversial.

7 Algebraic Forms of the Intrusive Methods

7.1 Reduced Basis Element Method (RBEM)

The algebraic form of RBEM for Ω_m is obtained by substituting the approximated solution and Lagrange multipliers into Eqs. 7 and 8, which is

$$\underline{\underline{M_m}}\dot{\mathbf{a}}_m + \underline{\underline{C_m}}\mathbf{a}_m + \underline{\underline{D_m}}\mathbf{q}_m = \underline{\underline{F_m}}\,\underline{\underline{Q_m}}\mathbf{a}_m - \underline{\underline{Q_n}}\mathbf{a}_n = 0. \tag{38}$$

The unknown coefficients of the RBs and Lagrange multipliers are collected in the vectors $\mathbf{a}_m$ and $\mathbf{q}_m$, respectively. $\mathbf{a}_n$ is the vector of coefficients for the neighboring subdomain Ω_n. They are defined as

$$\mathbf{a}_m = \left[a_{m,1}, \cdots, a_{m,N_{\mathrm{RB},m}}\right]^T, \quad \mathbf{q}_m = \left[\mathbf{q}_{m,1}, \cdots, \mathbf{q}_{m,N_{\lambda,[mn]}}\right]^T,$$
$$\mathbf{a}_n = \left[a_{n,1}, \cdots, a_{n,N_{\mathrm{RB},n}}\right]^T.$$

We also define the matrices. $\underline{\underline{M_m}}$[28] and $\underline{\underline{C_m}}$ represent the volumetric contributions, and their elements are computed as

$$M_{m,ij} = (v_{m,j}, w_{m,i}), \quad C_{m,ij} = \underline{\mathcal{Q}}_{\Omega}(v_{m,j}, w_{m,i}),$$

---

[28] We use the double underline to denote matrices and distinguish them from vectors.

with dimensions of $N_{\mathrm{RB},m} \times N_{\mathrm{RB},m}$, where $N_{\mathrm{RB},m}$ is the number of RBs for Ω_m.

$\underline{\underline{F_m}}$ is the source term with dimension $N_{\mathrm{RB},m} \times 1$, and its elements are

$$F_{m,i} = \underline{\mathcal{F}}(w_{m,i}) = (f, w_{m,i}).$$

$\underline{\underline{D_m}}$ and $\underline{\underline{Q_m}}$ correspond to the interface contributions. Consequently, they comprise a set of sub-blocks respective to each interface of Ω_m, i.e.,

$$\underline{\underline{D_m}} = \left\{ \underline{\underline{D_{[mn]}}} \middle| n \in N_\Gamma(m) \right\}, \quad \underline{\underline{Q_m}} = \left\{ \underline{\underline{Q_{[mn]}}} \middle| n \in N_\Gamma(m) \right\}.$$

Their elements are computed as

$$D_{[mn],ij} = \int_{\Gamma_{[mn]}} \zeta_{[mn],j}\ w_{m,i}, \quad Q_{[mn],ij} = \int_{\Gamma_{[mn]}} v_{m,j}\zeta_{[mn],i} - \int_{\Gamma_{[mn]}} v_{n,j}\zeta_{[mn],i},$$

with dimensions of $N_{\mathrm{RB},m} \times N_{\lambda,[mn]}$ and $N_{\lambda,[mn]} \times N_{\mathrm{RB},m}$, respectively.

Then, by assembling the equations of all subdomains, we can get the global system. The unknowns of the system are the coefficients of RBs and Lagrange multipliers for all subdomains and interfaces. The dimension of the system is the sum of the number of RBs for all subdomains and the number of Lagrange multipliers for all interfaces, which is

$$\sum_{m=1}^{N_\Omega} N_{\mathrm{RB},m} + \sum_{m=1}^{N_\Omega} \sum_{n \in N_\Gamma(m)} N_{\lambda,[mn]}$$

The global matrix can be expressed as

$$\underline{\underline{M}}\dot{\mathbf{a}} + \underline{\underline{C}}\mathbf{a} + \underline{\underline{D}}\mathbf{q} = \underline{\underline{F}}, \ \underline{\underline{Q}}a = 0,$$

where

$$\mathbf{a} = \begin{bmatrix} \mathbf{a}_1^T & \cdots & \mathbf{a}_{N_\Omega}^T \end{bmatrix}^T, \quad \mathbf{a}_m = \left[a_{m,1}, \cdots, a_{m,N_{\mathrm{RB},m}}\right]^T,$$
$$\text{and } \mathbf{q} = \begin{bmatrix} \mathbf{q}_1^T & \cdots & \mathbf{q}_{N_\Gamma}^T \end{bmatrix}^T, \quad \mathbf{q}_n = \left[q_{[mn],1}, \cdots, q_{[mn],N_{\lambda,[mn]}}\right]^T,$$

are the global vectors of coefficients of all local RBs and Lagrange multipliers, respectively. $\underline{\underline{M}}$, $\underline{\underline{C}}$, $\underline{\underline{D}}$, and $\underline{\underline{Q}}$ are the global matrices corresponding to the volumetric contributions and interface contributions. $\underline{\underline{F}}$ is the global source term. The elements of these matrices are assembled from the local contributions.

$\underline{\underline{M}}$, $\underline{\underline{C}}$, and $\underline{\underline{D}}$ are diagonal block matrices, namely,

$$\underline{\underline{M}} = \mathrm{diag}\left(\underline{\underline{M}}_1, \cdots, \underline{\underline{M}}_m\right), \ \underline{\underline{C}} = \mathrm{diag}\left(\underline{\underline{C}}_1, \cdots, \underline{\underline{C}}_m\right),$$
$$\underline{\underline{D}} = \mathrm{diag}\left(\underline{\underline{D}}_1, \cdots, \underline{\underline{D}}_m\right),$$

$\underline{\underline{Q}}$ represents the coupling between subdomains, and it is a sparse matrix with non-zero blocks corresponding to the interfaces. If we consider a row of $\underline{\underline{Q}}$ corresponding to the interface $\Gamma_{[mn]}$, the non-zero blocks are $\underline{\underline{Q_{[mn]}}}$ and $\underline{\underline{Q_{[nm]}}}$, and the rest of the blocks in the row are zero. This denotes

$$\underline{\underline{Q}} = \begin{bmatrix} \vdots & \vdots & \vdots & \vdots & \vdots \\ \cdots & \underline{\underline{Q_{[mn]}}} & \cdots & \underline{\underline{Q_{[nm]}}} & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix}.$$

See more details about the structure of the global matrix in [33].

7.2 Reduced Basis, Domain Decomposition, and Finite Element Method (RDF)

The algebraic form of the RDF is obtained considering the test functions in the internal and interface spaces, respectively. The algebraic form derived from Eq. 12 using the internal test functions $v_{m,i}$ is expressed as

$$\underline{\underline{M_m}}\dot{\mathbf{a}}_m + \underline{\underline{N_m}}\dot{\mathbf{b}}_m + \underline{\underline{C_m}}\mathbf{a}_m + \underline{\underline{D_m}}\mathbf{b}_m = \underline{\underline{F_m}}, \tag{39}$$

where $\mathbf{a}_m = [a_{m,1}, \cdots, a_{m,N_{\mathrm{RB},m}}]^T$ and $\mathbf{b}_m = [b_{m,1}, \cdots, b_{m,N_r}]^T$ are the vectors of coefficients of the internal and interface RBs, respectively. The matrices are defined as

$$M_{m,ij} = (v_{m,j}, v_{m,i}), N_{m,il} = (\phi_{r,l}, v_{m,i}), C_{m,ij} = \underline{\mathcal{Q}}_\Omega(v_{m,j}, v_{m,i})$$
$$+ \underline{\mathcal{Q}}_\Gamma(v_{m,j}, v_{m,i}), D_{m,il} = \underline{\mathcal{Q}}_\Gamma(\phi_{r,l}, v_{m,i}), F_{m,i} = \underline{\mathcal{F}}_m(v_{m,i}).$$

The algebraic form of Eq. 13 adopting the interface finite element functions $\phi_{r,l}$ is expressed as

$$\begin{aligned} &\underline{\underline{K_m}}\dot{\mathbf{a}}_m + \underline{\underline{H_m}}\dot{\mathbf{b}}_m + \underline{\underline{Q_m}}\mathbf{a}_m + \underline{\underline{G_m}}\mathbf{b}_m + \underline{\underline{K_n}}\dot{\mathbf{a}}_n \\ &+ \underline{\underline{H_n}}\dot{\mathbf{b}}_n + \underline{\underline{Q_n}}\mathbf{a}_n + \underline{\underline{G_n}}\mathbf{b}_n = \underline{\underline{F_m}} + \underline{\underline{F_n}}, \end{aligned} \tag{40}$$

where the vectors and matrices are defined as

$$\mathbf{a}_m = [a_{m,1}, \cdots, a_{m,N_{\mathrm{RB},m}}]^T, \quad \mathbf{b}_m = [b_{r,1}, \cdots, b_{r,N_r}]^T,$$
$$K_{m,lj} = (v_{m,j}, \phi_{r,l})\big|_{\Gamma_r}, \quad H_{m,lk} = (\phi_{r,k}, \phi_{r,l})\big|_{\Gamma_r}, \quad Q_{m,lj} = \underline{\mathcal{Q}}_\Gamma(v_{m,j}, \phi_{r,l})\big|_{\Gamma_r},$$
$$G_{m,lk} = \underline{\mathcal{Q}}_\Gamma(\phi_{r,k}, \phi_{r,l})\big|_{\Gamma_r}, \quad F_{m,l} = \underline{\mathcal{F}}_m(\phi_{r,l})\big|_{\Gamma_r}.$$

The vectors and matrices with subscript n are defined similarly, but with the test function $v_{n,j}$ defined on Ω_n.

Finally, the Eqs. A39 and A40 are assembled to obtain the global system, which is written as

$$\begin{aligned} \underline{\underline{M}}\dot{\mathbf{a}} + \underline{\underline{N}}\dot{\mathbf{b}} + \underline{\underline{C}}\mathbf{a} + \underline{\underline{D}}\mathbf{b} &= \underline{\underline{F}}_{\Omega}, \\ \underline{\underline{K}}\dot{\mathbf{a}} + \underline{\underline{H}}\dot{\mathbf{b}} + \underline{\underline{Q}}\mathbf{a} + \underline{\underline{G}}\mathbf{b} &= \underline{\underline{F}}_{\Gamma}. \end{aligned} \tag{41}$$

Let us clarify the structure of the global system. The unknowns of the system are the coefficients of the internal and interface RBs for all subdomains and interfaces, which are collected in the vectors $\mathbf{a} = [\mathbf{a}_1^{\mathrm{T}} \cdots \mathbf{a}_{N_\Omega}^{\mathrm{T}}]^T$ and $\mathbf{b} = [\mathbf{b}_1^{\mathrm{T}} \cdots \mathbf{b}_{N_\Gamma}^{\mathrm{T}}]^T$, respectively. The matrices $\underline{\underline{M}}$, $\underline{\underline{N}}$, $\underline{\underline{C}}$, and $\underline{\underline{D}}$ are block diagonal matrices consisting of the contributions inside each subdomain. For instance, $\underline{\underline{M}} = \operatorname{diag}(M_1, \cdots, M_m)$, and $M_{m,ij} = (v_{m,j}, v_{m,i})$. The other three matrices are expressed similarly as the inner product of v_i and/or ϕ_l, but are not detailed here for brevity.

In contrast, $\underline{\underline{K}}$, $\underline{\underline{H}}$, $\underline{\underline{Q}}$ and $\underline{\underline{G}}$ are sparse matrices that represent the coupling between subdomains. For example, $\underline{\underline{K}}$ has non-zero blocks for partitions sharing the same interface. If we consider a row of $\underline{\underline{K}}$ corresponding to the interface $\Gamma_r = \Omega_m \cap \Omega_n$, the non-zero blocks are $K_{m,lj} = (v_{m,j}, \phi_{r,l})|_{\Gamma_r}$ and $K_{n,lj} = (v_{n,j}, \phi_{r,l})|_{\Gamma_r}$, and the rest of the blocks in the row are zero. Thus, the structure of $\underline{\underline{K}}$ is

$$\underline{\underline{K}} = \begin{bmatrix} \vdots & \vdots & \vdots & \vdots & \vdots \\ \cdots & K_m & \cdots & K_n & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix}.$$

The other three matrices are stacked similarly, but with different definitions of the elements.

7.3 Port Reduction Reduced Basis Element Method

Similar to the above methods, the global system of the port reduction RBEM is obtained by composing the local systems of all subdomains and interfaces. The local formulations, Eq. 16, is converted to algebraic form by substituting the approximated solution into the equations, which is

$$\begin{aligned} \underline{\underline{M}}\dot{\mathbf{a}} + \underline{\underline{C}}\mathbf{a} + \underline{\underline{D}}\mathbf{b} &= \underline{\underline{F}}_{\Omega}, \\ \underline{\underline{H}}\dot{\mathbf{b}} + \underline{\underline{G}}\mathbf{b} &= \underline{\underline{F}}_{\Gamma}, \end{aligned} \tag{42}$$

where the unknowns are

$\mathbf{a} = [\mathbf{a}_1^{\mathrm{T}} \cdots \mathbf{a}_{N_\Omega}^{\mathrm{T}}]^T$ with $\mathbf{a}_m = [a_{m,1}, \cdots, a_{m,N_{\mathrm{RB},m}}]^T$, and
$\mathbf{b} = [\mathbf{b}_1^{\mathrm{T}} \cdots \mathbf{b}_{N_\Gamma}^{\mathrm{T}}]^T$ with $\mathbf{b}_r = [b_{r,1}, \cdots, b_{r,N_r}]^T$.

The matrices $\underline{\underline{M}}$, $\underline{\underline{C}}$, $\underline{\underline{D}}$, $\underline{\underline{H}}$, and $\underline{\underline{G}}$ are defined similarly as in the RDF method (Eq. A41) using bubble and port spaces defined in Eqs. 14 and 15, respectively. The main difference is that the trial space of SCRBEM includes the lifting functions, while the one of RDF does not. This leads to the fact that the coupling between internal and interface DoFs is only present in the first equation of Eq. A42, while in RDF, the coupling is present in both equations of Eq. A41.

7.4 Static Condensation Reduced Basis Element Method (SCRBEM)

Let us now explain the implementation of SCRBEM proposed by Huynh et al. in [198, 200]. They derived the method for the general *steady state* PDEs,

$$\underline{\mathcal{Q}}_{\Omega}(u, v; \boldsymbol{\mu}_j) + \underline{\mathcal{Q}}_{\Gamma}(u, v; \boldsymbol{\mu}_j) = \underline{\mathcal{F}}(v; \boldsymbol{\mu}_j), \quad v \in \mathcal{W}, \tag{A43}$$

to simplify the analysis, but readers should keep in mind that the procedure also holds for time-dependent scenarios.

The framework of *static condensation* involves bubble and port spaces. They are computed in archetypes, and then instantiated for the corresponding subdomains and interfaces. The following steps will illustrate how the *localized training* (see Sect. 3.3.3) is performed to compute the bubble and port spaces in the archetype domains and interfaces, respectively.

(1) First, we define a space $\hat{\mathcal{W}}_0^k = \operatorname{span}\{w \in H^1(\hat{\Omega}^k) : w|_{\partial\hat{\Omega}^k} = 0\}$. The bubble space in the archetype $\hat{\Omega}^k$ is the solution of Eq. A43. That is, finding $\hat{u}_0^k(\boldsymbol{\mu}) \in \hat{\mathcal{W}}_0^k$ such that

$$\underline{\mathcal{Q}}_{\Omega}(\hat{u}_0^k, \hat{w}_i^k; \boldsymbol{\mu}) = \underline{\mathcal{F}}(\hat{w}_i^k; \boldsymbol{\mu}) \quad \hat{w}_i^k \in \hat{\mathcal{W}}_0^k. \tag{44}$$

As $\hat{w}_0^k|_{\partial\hat{\Omega}^k} = 0$, $\underline{\mathcal{Q}}_{\Gamma}(\hat{u}_0^k, \hat{w}_i^k; \boldsymbol{\mu}) = 0$ is eliminated. If we have N_μ parameters, we will obtain the dataset $\{\hat{u}_0^k(\boldsymbol{\mu}_j)\}_{j=1}^{N_\mu}$. Then, the reduced bubble space $\hat{\mathcal{V}}_0^k \subset H_0^1(\hat{\Omega}^k)$ can be constructed by applying a dimensionality reduction, e.g., POD.

(2) The next stage is to find port bases that account for local boundary contributions. Consider an archetype interface $\hat{\Gamma}^e \in \partial\hat{\Omega}^k$, and the corresponding port modes are obtained by satisfying a boundary value problem. In [200], the Laplace equation is utilized, and the port functions $\hat{\psi}_j^e$ can be computed by solving

$$\int_{\hat{\Omega}^k} \nabla\hat{\psi}_j^e \cdot \nabla\hat{w}_0^k = 0, \quad \hat{w}_0^k \in \hat{\mathcal{W}}_0^k$$

with Dirichlet BCs defined as,

$$\hat{\psi}_j^e = \hat{\chi}_j^e, \qquad \text{on } \hat{\Gamma}^e,$$
$$\hat{\psi}_j^e = 0, \qquad \text{on } \partial\hat{\Omega}^k \setminus \hat{\Gamma}^e,$$

where $\hat{\chi}_j^e, j = 1, \ldots, \hat{N}_\Gamma^e$ are the eigenfunctions of the Laplace operator[29] defined on $\hat{\Gamma}^e$. The number of $\hat{\psi}_j^e$ and $\hat{\chi}_j^e$ is equal, and both are $\hat{N}_\Gamma^e$. We define $\hat{\Psi}^e = \text{span}\{\hat{\psi}_j^e\}_{j=1}^{\hat{N}_\Gamma^e} \subset H^1(\hat{\Omega}^k)$ as the port space for the archetype interface $\hat{\Gamma}^e$.

(3) Then, we solve a series of subproblems defined in $\hat{\Omega}^k$ with respect to the port functions $\hat{\psi}_j^e$. That is finding $\hat{\phi}_{0,j}^e \in \hat{\mathcal{V}}_0^k$,

$$\underline{\underline{\mathcal{Q}}}_\Omega(\hat{\phi}_{0,j}^e, \hat{w}_0^k; \boldsymbol{\mu}) = -\underline{\underline{\mathcal{Q}}}_\Omega(\hat{\psi}_j^e, \hat{w}_0^k; \boldsymbol{\mu}) \quad \hat{w}_0^k \in \hat{\mathcal{W}}_0^k.$$

Hence, for each parameter, we obtain $\hat{N}_\Gamma^e$ solutions $\hat{\phi}_{0,j}^e$. Note that the right-hand side of the above equation is known since $\hat{\psi}_j^e$ is computed in the previous step. We define another port space $\hat{\Phi}^e = \text{span}\{\hat{\phi}_{0,j}^e\}_{j=1}^{\hat{N}_\Gamma^e \times N_\mu} \subset H^1(\hat{\Omega}^k)$.

Through the above localized training approach, we obtain bubble space $\hat{\mathcal{V}}_0^k$ and port spaces $\hat{\Theta}^e$, which are written as

$$\begin{aligned} \hat{\mathcal{V}}_0^k &= \text{span}\{\hat{u}_0^k(\boldsymbol{\mu}_j)\}_{j=1}^{N_\mu}, \text{and } \hat{\Theta}^e = \hat{\Phi}^e \oplus \hat{\Psi}^e \\ &= \text{span}\{\hat{\phi}_{0,j}^e\}_{j=1}^{\hat{N}_\Gamma^e \times N_\mu} \oplus \text{span}\{\hat{\psi}_j^e\}_{j=1}^{\hat{N}_\Gamma^e}. \end{aligned} \tag{45}$$

Recall that $\hat{\mathcal{V}}_0^k$ vanishes on the whole boundary of $\hat{\Omega}^k$, while $\hat{\Theta}^e$ is defined in both the interior and the surface.

Next, we will demonstrate how archetype spaces are mapped to their respective subdomains and interfaces, as well as how static condensation is performed.

The computational domain Ω is decomposed into N_Ω subdomains Ω_m, and the internal faces are Γ_r. They are instantiated from the archetype $\hat{\Omega}^k$ and $\hat{\Gamma}^e$, respectively, by applying the corresponding geometric transformations. Thus, $\Omega_m = \underline{\underline{T}}_\Omega^m \circ \hat{\Omega}^k$ and $\Gamma_r = \underline{\underline{T}}_\Gamma^r \circ \hat{\Gamma}^e$, where $\underline{\underline{T}}_\Omega^m$ and $\underline{\underline{T}}_\Gamma^r$ are the geometric transformations for Ω_m and Γ_r, respectively.

Similarly, the bubble and port spaces in the archetype are instantiated for the corresponding subdomains and interfaces, which are denoted by $\mathcal{V}_{m,0} = \underline{\underline{T}}_\Omega^m \circ \hat{\mathcal{V}}_0^k$ and $\Theta_r = \underline{\underline{T}}_\Gamma^r \circ \hat{\Theta}_e$. Note that $\mathcal{V}_{m,0}$ only defines the basis in the interior of Ω_m, while Θ_r is defined in both the interior and the surface, which is different from Φ_r given in Eq. 14.

The local solution in Ω_m can be approximated by

$$\begin{aligned} u_m(\boldsymbol{\mu}) &\approx \sum_{i=1}^{N_m} a_{m,i}(\boldsymbol{\mu}) v_{m,i} \\ &+ \sum_{\Gamma_r \in \mathcal{I}_m} \sum_{j=1}^{N_r} b_{r,j}(\boldsymbol{\mu}) \theta_{r,j}, \\ &\text{with } v_{m,i} \in \mathcal{V}_{m,0} \text{and } \theta_{r,j} \in \Theta_r, \end{aligned}$$

where $a_{m,i}$ and $b_{r,j}$ are the unknown coefficients of the system, $\mathcal{I}_m$ is the set of interfaces connected to Ω_m, N_m and N_r are the dimensions of the bubble and port spaces, respectively.

Then, we project the steady system A43 onto the bubble and port spaces (Eq. A45), which yields

$$\begin{aligned} &\underline{\underline{\mathcal{Q}}}_\Omega\Big(\sum_{i=1}^{N_{\text{RB},m}} a_{m,i} v_{m,i} + \sum_{i=1}^{N_r^p} b_{r,i}\theta_{r,i}, v_{m,i}\Big) \\ &= \underline{\underline{\mathcal{F}}}_m(v_{m,i}) \qquad v_{m,i} \in \mathcal{V}_{m,0}, \\ &\underline{\underline{\mathcal{Q}}}_\Omega\Big(\sum_{i=1}^{N_{\text{RB},m}} a_{m,i} v_{m,i} + \sum_{i=1}^{N_r^p} b_{r,i}\theta_{r,i}, \theta_{r,i}\Big) \\ &+ \underline{\underline{\mathcal{Q}}}_\Gamma\Big(\sum_{i=1}^{N_r^p} b_{r,i}\theta_{r,i}, \theta_{r,i}\Big) = \underline{\underline{\mathcal{F}}}_m(\theta_{r,i}) \quad \theta_{r,i} \in \Theta_r. \end{aligned}$$

Note that $\boldsymbol{\mu}$ is omitted for brevity.

The algebraic form of the above system is written as

$$\begin{aligned} \underline{\underline{C}}\mathbf{a} + \underline{\underline{D}}\mathbf{b} &= \underline{\underline{F}}_\Omega, \\ \underline{\underline{Q}}\mathbf{a} + \underline{\underline{G}}\mathbf{b} &= \underline{\underline{F}}_\Gamma, \end{aligned}$$

and the *static condensation* is performed by eliminating $\mathbf{a}$, which leads to

$$(\underline{\underline{G}} - \underline{\underline{Q}}\,\underline{\underline{C}}^{-1}\underline{\underline{D}})\mathbf{b} = \underline{\underline{F}}_\Gamma - \underline{\underline{Q}}\,\underline{\underline{C}}^{-1}\underline{\underline{F}}_\Omega,$$

where the unknowns are $\mathbf{b} = [\mathbf{b}_1^{\text{T}} \cdots \mathbf{b}_{N_\Gamma}^{\text{T}}]^T$, with $\mathbf{b}_r = [b_{r,1}, \cdots, b_{r,N_r}]^T$. The matrices $\underline{\underline{C}}$, $\underline{\underline{D}}$, $\underline{\underline{Q}}$, and $\underline{\underline{G}}$ are defined similarly as in the RDF method (Eq. A41) using trial and test spaces defined in Eqs. 45.

8 Mathematical Formulation for Preliminary Techniques

In this section, we will present the mathematical formulation of some preliminary techniques, which are used in the main text. The techniques include the active subspace method, the Partition of Unity method, proper generalized decomposition, variational interpretation of the embedded

---

[29] (see more about eigenfunctions of the Laplace operator in [53]).

boundary method, and the illustration for several widely used neural network architectures.

8.1 Active Subspace

Here, we will only present the steps of *AS* technique, whilst for more theoretical and mathematical details, as well as applications, we refer readers to [127].

We consider a multi-dimensional parameter space $\mathcal{P} \subset \mathbb{R}^{N_{\mathcal{P}}}$. We also regard the parametric function

$\underline{f}(\boldsymbol{\mu}) : \mathbb{R}^{N_{\mathcal{P}}} \rightarrow \mathbb{R}$. The uncertainty in the input parameters is accounted for with a probability density function ρ.

The gradient of $\underline{f}$ with respect to $\boldsymbol{\mu}$ is used, namely, $\nabla_{\boldsymbol{\mu}}\underline{f}(\boldsymbol{\mu}) \in \mathbb{R}^{N_{\mathcal{P}}}$. We form its non-centered covariance matrix $\underline{\underline{\Sigma}}$,

$$\underline{\underline{\Sigma}} = \mathbb{E}\left[\nabla_{\boldsymbol{\mu}}\underline{f}(\boldsymbol{\mu})\nabla_{\boldsymbol{\mu}}\underline{f}(\boldsymbol{\mu})^{\mathrm{T}}\right] = \int \left(\nabla_{\boldsymbol{\mu}}\underline{f}(\boldsymbol{\mu})\right)\left(\nabla_{\boldsymbol{\mu}}\underline{f}(\boldsymbol{\mu})\right)^{\mathrm{T}} \rho \, \mathrm{d}\boldsymbol{\mu},$$

where $\mathbb{E}$ is the expected value. Matrix $\underline{\underline{\Sigma}}$ is symmetric and positive semidefinite. Therefore, it has a real eigenvalue decomposition,

$$\underline{\underline{\Sigma}} = \underline{\underline{W}}\,\underline{\underline{\Lambda}}\,\underline{\underline{W}}^{\mathrm{T}},$$

where $\underline{\underline{W}}$ is orthogonal and contains the eigenvectors in columns. $\underline{\underline{\Lambda}} = \mathrm{diag}\{\lambda_1, \ldots, \lambda_{N_{\mathcal{P}}}\}$ with the eigenvalues $\lambda_1 \geq \cdots \lambda_{N_{\mathcal{P}}} \geq 0$.

A key aspect of the method consists of realizing that eigenvalues and eigenvectors can be partitioned

$$\underline{\underline{\Lambda}} = \begin{bmatrix} \underline{\underline{\Lambda}}_1 & \\ & \underline{\underline{\Lambda}}_2 \end{bmatrix}, \quad \underline{\underline{W}} = \left[\underline{\underline{W}}_1, \underline{\underline{W}}_2\right].$$

Note that $\underline{\underline{W}}_1$ with a shape of $N_{\mathcal{P}} \times M$ denotes the M-dimension *active* subspace of $\underline{\underline{W}}$. $\underline{\underline{W}}_1$ captures the most of the variation of $\nabla_{\boldsymbol{\mu}}\underline{f}(\boldsymbol{\mu})$. Then, the low-dimensional *active* parameter can be defined as a linear combination of the original values, indeed, $\boldsymbol{\mu}_M = \underline{\underline{W}}_1^{\mathrm{T}}\boldsymbol{\mu}$. With the active variables, we can approximate the function of interest $\underline{f}$,

$$\underline{f}(\cdot, \boldsymbol{\mu}) \approx \underline{g}(\cdot, \underline{\underline{W}}_1^{\mathrm{T}}\boldsymbol{\mu}) = \underline{g}(\cdot, \boldsymbol{\mu}_M).$$

8.2 An Example of the Partition of Unity (PoU) Method

Here, we present a linear piecewise formulation for a 1D problem as an example. The PoU functions and the resulting global basis are sketched in Fig. 34. To satisfy the above requirements, the values of the PoU function in the local coordinate are given as

$$\underline{\psi}_m(x) = \begin{cases} \dfrac{x}{\delta_1}, & 0 \leq x \leq \delta_1 \\ 1, & \delta_1 < x < h - \delta_2, \\ -\dfrac{x}{\delta_2} + \dfrac{h}{\delta_2}, & h - \delta_2 \leq x \leq h, \end{cases}$$

where the local coordinate of a partition is assigned in the range of $x \in [0, h]$, and $0 \leq x \leq \delta_1$ and $h - \delta_2 \leq x \leq h$ are overlapping regions. Note that the same $\underline{\psi}$ was defined here for all partitions Ω_m. But this is not necessary, and different $\underline{\psi}_m$ can be defined for each Ω_m. Even more, multiple $\phi_{m,i}$ can be defined in each Ω_m and mode $v_{m,i}$ [248].

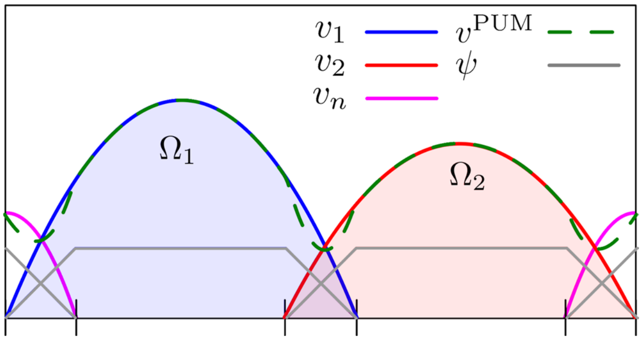


Fig. 34 Sketch of PoU for a one-dimensional (1D) case showing two subdomains Ω_1 and Ω_2. Their first basis functions $v_{1,1}$ and $v_{2,1}$ are marked in blue and red, respectively. The pink lines are vectors for segments out of the picture. The PoU function $\underline{\psi}$ is displayed with a gray line in each subdomain. The vector resulting from the PUM procedure, v_i^{PUM}, is shown in green

8.3 Proper Generalized Decomposition (PGD) for Parametric Problems

To better illustrate the PGD method, we will explain the procedure proposed by Discacciati et al. in [55], which is a PGD-based domain decomposition method for parametric problems. The authors divided the parameters into two sets, $\boldsymbol{\mu}_{\mathrm{BC}}$ and $\boldsymbol{\mu}$, which are corresponding to the interface conditions and the rest of the problem, respectively. The former is used to parameterize the conditions of subdomains to generate local snapshots, while the latter is for the rest of the problem.

The implementation of PGD in [55] consists of several steps. First, the local solution is divided into two parts

$$u_m(\mathbf{x}; \boldsymbol{\mu}, \boldsymbol{\mu}_{\mathrm{BC}}) = v_{m,0}(\mathbf{x}; \boldsymbol{\mu}) + \tilde{v}_m(\mathbf{x}; \boldsymbol{\mu}, \boldsymbol{\mu}_{\mathrm{BC}}),$$

where $v_{m,0}(\mathbf{x}, \boldsymbol{\mu})$ denotes the local solution satisfying homogeneous interface conditions, global boundary conditions, and source terms. $\tilde{v}_m(\mathbf{x}; \boldsymbol{\mu}, \boldsymbol{\mu}_{\mathrm{BC}})$ only represents the interface contribution without any sources and boundary conditions. The solution is parameterized with $\boldsymbol{\mu}$ and $\boldsymbol{\mu}_{\mathrm{BC}}$. The latter is the parameter vector for the interface conditions, and the former is for the rest of the problem. $\boldsymbol{\mu}_{\mathrm{BC}}$ is used to parameterize the conditions of subdomains to generate local snapshots.

Then, $\tilde{v}_m(\mathbf{x}; \boldsymbol{\mu}, \boldsymbol{\mu}_{\mathrm{BC}})$ is further split into two items, namely,

$$\tilde{v}_m(\mathbf{x}; \boldsymbol{\mu}, \boldsymbol{\mu}_{\mathrm{BC}}) = v_m(\mathbf{x}; \boldsymbol{\mu}, \boldsymbol{\mu}_{\mathrm{BC}}) + \sum_{\Gamma_r \in \mathcal{I}_m} \sum_{q=1}^{N_{\Gamma_r}} \Lambda_q^{\Gamma_r}(\boldsymbol{\mu}_{\mathrm{BC}}) \eta_q^{\Gamma_r}(\mathbf{x}),$$

where $\mathcal{I}_m$ is the set of interfaces associated with Ω_m. $\eta_q^{\Gamma_r}$ and $\Lambda_q^{\Gamma_r}$ are suitable basis functions and corresponding coefficients that are capable of approximating interface quantities.

Thirdly, the PGD approach is applied to approximate the two parts $v_{m,0}$ and v_m, which yields

$$v_{m,0}(\mathbf{x}; \boldsymbol{\mu}) = \sum_{i=1}^{N_{m,0}^{\mathrm{PGD}}} X_{m,0,i}(\mathbf{x}) M_{m,0,i}(\boldsymbol{\mu}), \text{ and}$$

$$v_m(\mathbf{x}; \boldsymbol{\mu}, \boldsymbol{\mu}_{\mathrm{BC}}) = \sum_{i=1}^{N_r^{\mathrm{PGD}}} X_{r,i}(\mathbf{x}) M_{r,i}(\boldsymbol{\mu}) K_{r,i}(\boldsymbol{\mu}_{\mathrm{BC}}).$$

Finally, the complete approximation is written by

$$\begin{aligned} u_m(\mathbf{x}; \boldsymbol{\mu}, \boldsymbol{\mu}_{\mathrm{BC}}) = & \sum_{i=1}^{N_{m,0}^{\mathrm{PGD}}} X_{m,0,i}(\mathbf{x}) M_{m,0,i}(\boldsymbol{\mu}) \\ & + \sum_{i=1}^{N_m^{\mathrm{PGD}}} X_{m,i}(\mathbf{x}) M_{m,i}(\boldsymbol{\mu}) K_{m,i}(\boldsymbol{\mu}_{\mathrm{BC}}) \\ & + \sum_{\Gamma_r \in \mathcal{I}_m} \sum_{q=1}^{N_{\Gamma_r}} \Lambda_q^{\Gamma_r}(\boldsymbol{\mu}_{\mathrm{BC}}) \eta_q^{\Gamma_r}(\mathbf{x}). \end{aligned}$$

Note that the algorithms for computing PGD modes and boundary basis functions fall outside the scope of this review. Readers may consult the original publications. We also recommend a book [359] for a comprehensive explanation of PGD.

8.4 Variational Interpretation of the Embedded Boundary Method

To further clarify the *embedded boundary method* operator $\underline{\mathcal{Q}}$ presented in the main text, we present the weak forms of a popular method for handling embedded boundaries: the *shifted boundary method*. The discussion is focused on a Poisson problem for simplicity, and interested readers can refer to the original publications for more details and other types of PDEs.

Consistent with the notation in the main text (as shown in Fig. 22), $\mathcal{B}$ denotes the background domain, $\mathcal{D}$ the physical subdomain, $\mathcal{D}_{\mathcal{T}}$ the surrogate domain. The background mesh is $\mathcal{B}_h$, embedded mesh is $\mathcal{T}_h$. The elements intersected by the boundary $\partial\mathcal{D}$ are called cut cells, noted as G_h.

The shifted boundary method [278–280] maps the boundary conditions from the true boundary $\partial\mathcal{D}$ to a surrogate boundary $\partial\mathcal{D}_{\mathcal{T}}$ (the faces of the background mesh closest to $\partial\mathcal{D}$). The value at the true boundary $\tilde{u}_d(\tilde{\mathbf{x}})$ is related to the value at the surrogate boundary $u_d(\mathbf{x})$ via a first-order Taylor expansion:

$$\tilde{u}_d(\tilde{\mathbf{x}}) \approx u_d(\mathbf{x}) + \nabla u_d(\mathbf{x}) \cdot \mathbf{d}$$

where $\tilde{\mathbf{x}} = \mathbf{d} + \mathbf{x}$ is the distance vector. $\tilde{\mathbf{x}}$ lies on $\partial\mathcal{D}$ and $\mathbf{x}$ lies on $\partial\mathcal{D}_{\mathcal{T}}$. The weak form on the surrogate domain $\mathcal{D}_{\mathcal{T}}$ seeks u_d such that:

$$\underline{\mathcal{Q}}(u_d, w)_{\mathcal{D}_{\mathcal{T}}} = \underline{\mathcal{F}}(w)$$

where the boundary operator $\underline{\mathcal{Q}}$ incorporates the Nitsche penalty and the Taylor shift:

$$\begin{aligned} \underline{\mathcal{Q}} = & (\nabla u_d, \nabla w)_{\mathcal{D}_{\mathcal{T}}} - (\nabla u_d \cdot \mathbf{n}, w + \nabla w \cdot \mathbf{d})_{\partial\mathcal{D}_{\mathcal{T}}} \\ & - \left(\nabla u_d \cdot \mathbf{d}, \frac{\mathbf{n} \cdot \mathbf{n}}{\|\mathbf{d}\|} \nabla w \cdot \mathbf{n} \right)_{\partial\mathcal{D}_{\mathcal{T}}} \\ & + \left(u_d + \nabla u_d \cdot \mathbf{d}, \frac{\alpha}{h}(w + \nabla w \cdot \mathbf{d}) \right)_{\partial\mathcal{D}_{\mathcal{T}}}, \end{aligned}$$

where $\tilde{n}$ is the normal vector of the surrogate boundary $\partial\mathcal{D}_{\mathcal{T}}$, and α is the penalty parameter. The left hand side term $\underline{\mathcal{F}}$ is defined as

$$\underline{\mathcal{F}}(w) = (f, w)_{\mathcal{D}_{\mathcal{T}}} - (g_D, \nabla w \cdot \mathbf{n})_{\partial\mathcal{D}_{\mathcal{T}}} - ((\nabla g_D \cdot \boldsymbol{\tau})\boldsymbol{\tau} \cdot n, \nabla w \cdot \mathbf{d}) + \left(g_D, \frac{\alpha}{h}(w + \nabla w \cdot \mathbf{d}) \right),$$

where g_D is the Dirichlet data on the true boundary $\partial\mathcal{D}$, $\boldsymbol{\tau}$ is the tangential vector of $\partial\mathcal{D}$, α is the Nitsche penalty parameter, h is a characteristic length of the elements in the direction orthogonal to the boundary. See more details in [279, 280], and the incorporation of the embedded framework with ROMs in [278].

The values in the rest region, $\mathcal{B} \setminus \mathcal{D}_{\mathcal{T}}$, are already given in Eq. 25, and no further discussion is needed here.

Be aware that the procedure can be easily extended to transient problems by considering the time stepping. The formulations for more complex phenomena, e.g., Stokes, linear advection-diffusion and incompressible Navier-Stokes equations are illustrated in [279, 280].

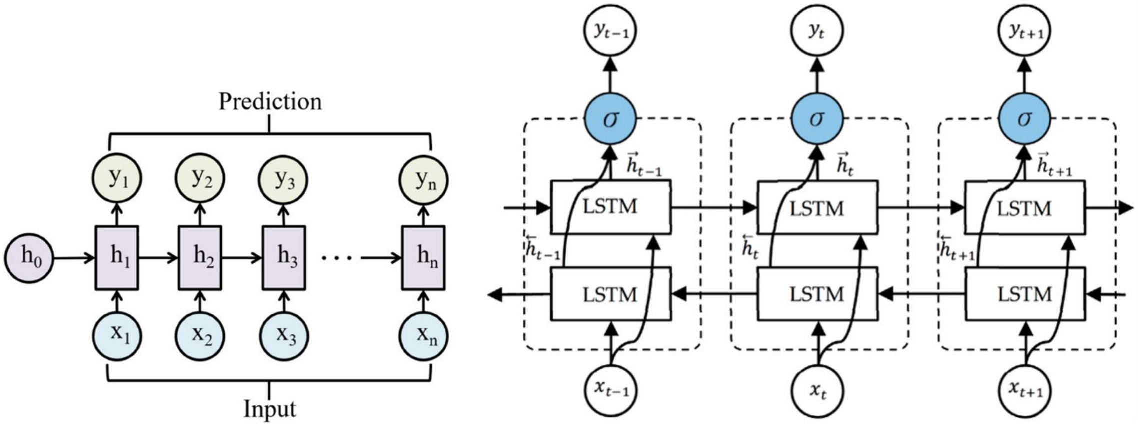


Fig. 35 Architectures of recurrent neural networks for a sequential dataset. *Left*: standard recurrent network with initial input layer h_0 and middle hidden nodes $h_1, \ldots, h_n$. Multiple inputs $x_1, \ldots, x_n$ and corresponding outputs $y_1, \ldots, y_n$. *Right*: long short-term memory networks: time-dependent inputs, x_{t-1}, x_t, x_{t+1}, and outputs y_{t-1}, y_t, y_{t+1}. σ is the activation function. $\vec{h}_t$ is the hidden node at time t. An additional path of nodes for feedback from the next sequence. Figures are taken with permission from [360], copyright owned by MDPI

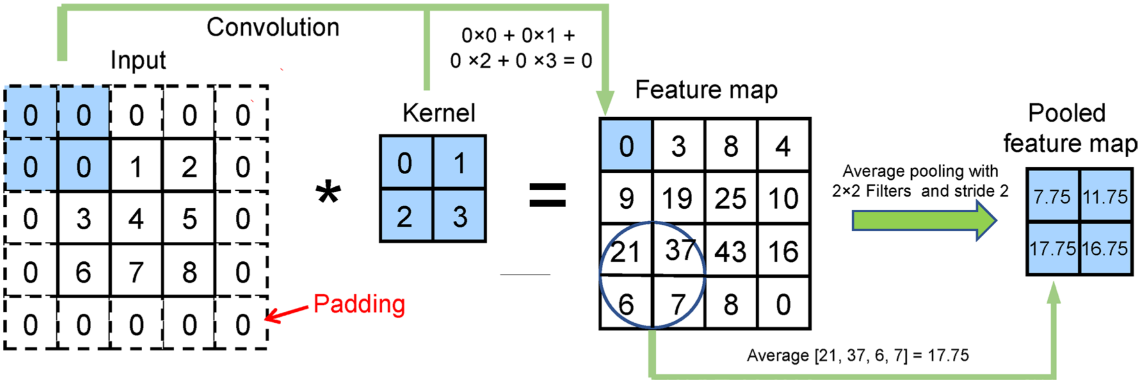


Fig. 36 Example of convolution and pooling layers in the convolutional neural network. Kernel is a window to weight the input data. Pooled map is used to reduce the dimensions. Figure taken with permission from [361], copyright owned by MDPI

8.5 Neural Network Architectures

The neural network can be classified by considering the exchange between neurons. For the ANN plotted in Fig. 24, the inputs are only transferred forward to reach the final outputs. However, for sequential data and transient phenomena, it requires results at different temporal points. This can be achieved by the so-called *recurrent neural networks*, in which the output of nodes can feed back into the model [15, 360]. It is also worth mentioning an advanced variant, *long short-term memory*, which can transfer information both forwards and backwards. The two are displayed in Fig. 35 for clarification.

Another widely used architecture is called *convolutional neural network*, which consists of special additional layers to extract data in subregions. These denote the convolution and pooling process shown in Fig. 36.

Other frameworks, such as *graph neural networks* [235, 362] and the neural operator [363, 364], are gaining popularity nowadays. They are mentioned here for completeness and without further explanation.

A widely used dimensionality reduction technique, *autoencoder*, is explained as follows. Its architecture is sketched in Fig. 37. The inputs are snapshots $u(\boldsymbol{\mu})$, and the dimensionality is reduced by the *encoder* into a low-dimensional *latent space* $\boldsymbol{\alpha}(\boldsymbol{\mu})$. By analogy with POD, $\boldsymbol{\alpha}(\boldsymbol{\mu})$ can be regarded as coefficients of a very low dimension. The reconstructed fields $u'(\boldsymbol{\mu})$ are obtained via the *decoder* operation. The framework is built to minimize the difference between $u(\boldsymbol{\mu})$ and $u'(\boldsymbol{\mu})$, and the loss function is given by $\mathcal{E} = \frac{1}{N_u}\|u(\boldsymbol{\mu}) - u'(\boldsymbol{\mu})\|_2^2$.

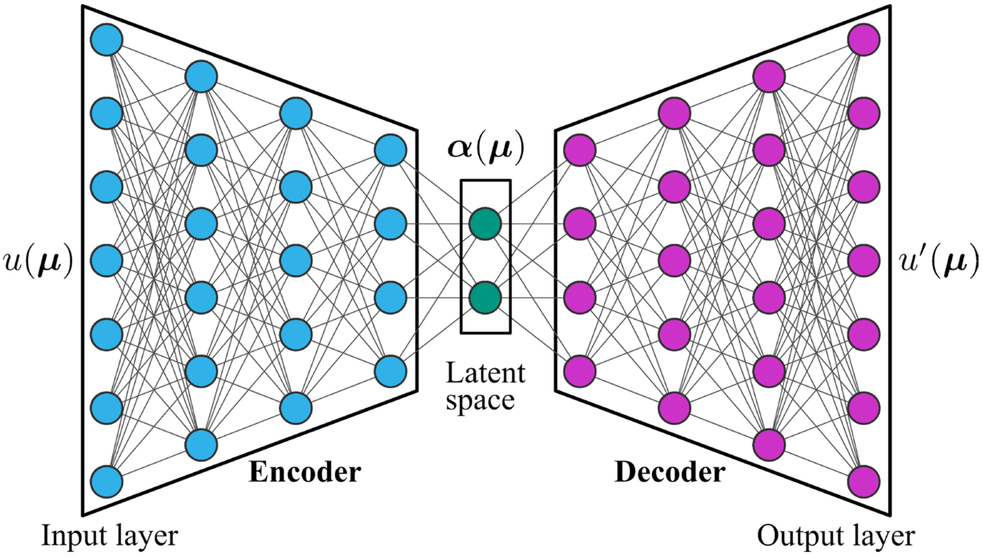


Fig. 37 A sketch of an *auto-encoder*. A high-fidelity scalar field $u(\boldsymbol{\mu})$ is compressed by the *encoder* into a low-dimensional *latent space* $\boldsymbol{\alpha}(\boldsymbol{\mu})$. The *decoder* is applied to reconstruct the solution $u'(\boldsymbol{\mu})$ from $\boldsymbol{\alpha}(\boldsymbol{\mu})$

Acknowledgements The authors acknowledge Dr. Jorge Yanez (KIT, the Karlsruhe Institute of Technology, Germany) for his help in improving the language used in this paper. They also thank the Institute for Thermal Energy Technology and Safety (ITES) and the Karlsruhe House of Young Scientists (KHYS) at KIT for funding Mr Shenhui Ruan's visit to the International School for Advanced Studies (SISSA), thereby facilitating collaboration. They appreciate the China Scholarship Council (CSC) for funding the doctoral study of Mr. Shenhui Ruan.

Author Contributions Shenhui Ruan did the literature review, read all references, categorized the existing studies, and wrote the main manuscript text. Andreas Class conceptualized this review and gave suggestions for draft revision. Gianluigi Rozza provided suggestions for references that should be included and the idea for the structure of this paper. All authors reviewed the manuscript.

Funding Open Access funding enabled and organized by Projekt DEAL. The authors received no specific grant, funding, or financial support from any public, commercial, or not-for-profit funding agencies for the research, authorship, and/or publication of this article.

Data Availability No datasets were generated or analysed during the current study.

Declarations

Competing Interests The authors declare no competing interests.

References

1. Roelofs F, Uitslag-Doolaard H, Dovizio D, Mikuz B, Shams A, Bertocchi F, Rohde M, Pacio J, Di Piazza I, Kennedy G, Van Tichelen K, Obabko A, Merzari E (2019) Towards validated prediction with RANS CFD of flow and heat transport in a wire-wrap fuel assembly. Nucl Eng Des 353:110273. https://doi.org/10.1016/j.nucengdes.2019.110273
2. Wu S, Wang D, Zhang Y, Okamoto K, Pellegrini M, Tian W, Qiu S, Su GH (2024) Model development for oxidation and degradation behavior of accident tolerant cr coating on zr alloy cladding under high temperature steam atmosphere. J Nucl Mater. 589:154836. https://doi.org/10.1016/j.jnucmat.2023.154836
3. Chen L, Jin S, Bu G (2017) Experimental investigation of a novel multi-tank thermal energy storage system for solar-powered air conditioning. Appl Therm Eng 123:953–962. https://doi.org/10.1016/j.applthermaleng.2017.05.160
4. Moukalled F, Mangani L, Marwan Darwish FM, Mangani L, Darwish M (2016) The finite volume method. Springer
5. Ern A, Guermond J-L (2004) Theory and practice of finite elements, vol 159. Springer
6. Claudio Canuto MYH, Quarteroni A, Zang TA (2007) Spectral methods: evolution to complex geometries and applications to fluid dynamics. Springer Science & Business Media
7. Zhong M, Xiao T, Krause MJ, Frank M, Simonis S (2024) A stochastic Galerkin lattice Boltzmann method for incompressible fluid flows with uncertainties. J Comput Phys. 517:113344. https://doi.org/10.1016/j.jcp.2024.113344
8. Kunisch K, Volkwein S (January 2002). Galerkin proper orthogonal decomposition methods for a general equation in fluid dynamics. SIAM J Numer Anal. 40(2):492–515. ISSN 0036-1429, 2323 1095-7170. https://doi.org/10.1137/S0036142900382612
9. Rozza G, Stabile G, Ballarin F (2022) Advanced reduced order methods and applications in computational fluid dynamics. SIAM
10. Bertrand F, Boffi D, Halim A (2023 February) A reduced order model for the finite element approximation of eigenvalue problems. Comput Methods Appl Mech Eng. 404:115696. ISSN 00457825. https://doi.org/10.1016/j.cma.2022.115696
11. Mirhoseini MA, Zahr MJ (2023 January) Model reduction of convection-dominated partial differential equations via optimization-based implicit feature tracking. J Comput Phys. 473:111739. ISSN 00219991. https://doi.org/10.1016/j.jcp.2022.111739
12. Quarteroni A, Manzoni A, Negri F (2015) Reduced basis methods for partial differential equations: an introduction, vol 92. Springer

13. Kunisch K, Volkwein S (January 2008) Proper orthogonal decomposition for optimality systems. Esaim: M2an 42(1):1–23. ISSN 0764-583X, 1290-3841. https://doi.org/10.1051/m2an:2007054
14. Hesthaven JS, Rozza G, Stamm B (2016) Certified reduced basis methods for parametrized partial differential equations, vol 590. Springer
15. Brunton SL, Kutz JN (2024) Data-driven science and engineering: machine learning, dynamical systems, and control. Cambridge University Press
16. Bai Z, Peng L (2021) Non-intrusive nonlinear model reduction via machine learning approximations to low-dimensional operators. Adv. Model. Simul. Eng. Sci 8(1):28. https://doi.org/10.1186/s40323-021-00213-5
17. Padula G, Girfoglio M, Rozza G (2024) A brief review of reduced order models using intrusive and non-intrusive techniques. arXiv preprint arXiv:2406.00559. https://doi.org/10.1002/pamm.202400210
18. Heaney CE, Wolffs Z, Atli Tómasson JA, Kahouadji L, Salinas P, Nicolle A, Navon IM, Matar OK, Srinil N, Pain CC (2022) An AI-based non-intrusive reduced-order model for extended domains applied to multiphase flow in pipes. Phys Fluids 34(5). https://doi.org/10.1063/5.0088070
19. Wentland CR, Rizzi F, Barnett J, Tezaur I (2025) The role of interface boundary conditions and sampling strategies for Schwarz-based coupling of projection-based reduced order models. J Comput Appl Math 465:116584. arXiv preprint arXiv:2410.04668. https://doi.org/10.1016/j.cam.2025.116584
20. Barone MF, Kalashnikova I, Segalman DJ, Thornquist HK (2009 April) Stable Galerkin reduced order models for linearized compressible flow. J Comput Phys. 228(6):1932–1946. https://doi.org/10.1016/j.jcp.2008.11.015
21. Kalashnikova I, Waanders BVB, Arunajatesan S, Barone M (2014 April) Stabilization of projection-based reduced order models for linear time-invariant systems via optimization-based eigenvalue reassignment. Comput Methods Appl Mech Eng. 272:251–270. (ISSN 00457825). https://doi.org/10.1016/j.cma.2014.01.011
22. Løvgren AE, Maday Y, Rønquist EM (2006) The reduced basis element method for fluid flows. Anal Simul Fluid Dyn. 129–154. Springer
23. Antonietti PF, Pacciarini P, Quarteroni A (2016) A discontinuous Galerkin reduced basis element method for elliptic problems. Esaim: M2an 50(2):337–360. https://doi.org/10.1051/m2an/2015045
24. Riffaud S, Bergmann M, Farhat C, Grimberg S, Iollo A (2021) The DGDD method for reduced-order modeling of conservation laws. J Comput Phys. 437:110336. https://doi.org/10.1016/j.jcp.2021.110336
25. Sambataro G (2022) Component-based model order reduction procedures for large scale THM systems. PhD thesis, Bordeaux
26. Batta A, Class AG (2017) CFD analysis of pressure drop across grid spacers in rod bundles compared to correlations and heavy liquid metal experimental data. Nucl Eng Des. 312:121–127. https://doi.org/10.1016/j.nucengdes.2016.08.040
27. Ruan S, Yanez J, Class AG (2024) Local reduced subspaces of subchannel-inspired subdomains. Numer Meth Eng. 125(18):e7552. https://doi.org/10.1002/nme.7552
28. Shihao W, Xiaocheng W, Zhang Y, Wang D, Pegfei M, Tian W, Qiu S, Su GH (2025) Study on eutectic melting and high-temperature oxidation characteristics of cr coated zr alloy cladding under steam atmosphere at 1400~ 1500°C. J Alloys Compd. 183389
29. Bühler L, Mistrangelo C (2024) Geometric optimization of electrically coupled liquid metal manifolds for WCLL blankets. IEEE Trans Plasma Sci. 52(9):3418–3423. https://doi.org/10.1109/TPS.2024.3362689
30. Bergman TL (2011) Fundamentals of heat and mass transfer. John Wiley & Sons
31. Fan L, Tu Z, Chan SH (2021) Recent development of hydrogen and fuel cell technologies: a review. Energy Rep 7:8421–8446. https://doi.org/10.1016/j.egyr.2021.08.003
32. Roos P, Haselbacher A (2021) Thermocline control through multi-tank thermal-energy storage systems. Appl Energy. 281:115971. https://doi.org/10.1016/j.apenergy.2020.115971
33. Pegolotti L, Pfaller MR, Marsden AL, Deparis S (2021) Model order reduction of flow based on a modular geometrical approximation of blood vessels. Comput Methods Appl Mech Eng. 380:113762. https://doi.org/10.1016/j.cma.2021.113762
34. Zappon E, Manzoni A, Gervasio P, Quarteroni A (2024) A reduced order model for domain decompositions with non-conforming interfaces. J Sci Comput. 99(1):22. https://doi.org/10.1007/s10915-024-02465-w
35. Cinquegrana D, Donelli RS, Viviani A (2011) A hybrid method based on POD and domain decomposition to compute the 2-D aerodynamics flow field. AIMETA, Bologna, Italy. 1–10
36. Barnett J, Tezaur I, Mota A (2022) The Schwarz alternating method for the seamless coupling of nonlinear reduced order models and full order models. arXiv preprint arXiv:2210.12551
37. Discacciati N, Hesthaven JS (2023) Localized model order reduction and domain decomposition methods for coupled heterogeneous systems. Numer Meth Eng. 124(18):3964–3996. https://doi.org/10.1002/nme.7295
38. Ivan P, Torlo D, Nonino M, Rozza G (2024) A time-adaptive algorithm for pressure dominated flows: a heuristic estimator. arXiv preprint arXiv:2407.00428
39. Zappon E, Manzoni A, Quarteroni A (2023 April) Efficient and certified solution of parametrized one-way coupled problems through DEIM-based data projection across non-conforming interfaces. Adv Comput Math 49(2):21. ISSN 1019-7168, 1572-9044. https://doi.org/10.1007/s10444-022-10008-w
40. Du P, Shan J, Zhang B, Leung LKH (2019) Thermal-hydraulics analysis of flow blockage events for fuel assembly in a sodium-cooled fast reactor. Int J Heat Mass Transf. 138:496–507. https://doi.org/10.1016/j.ijheatmasstransfer.2019.04.073
41. Todreas NE, Kazimi MS, Massoud M (2021) Nuclear systems volume II: elements of thermal hydraulic design. CRC Press
42. Class A, Viellieber M, Moussiopoulos N, Barmpas F (2014) Two way coupled micro/meso scale method for wind farms. Proc Appl Math Mech. 14(1):585–586. https://doi.org/10.1002/pamm.201410280
43. Viellieber M, Class A (2015) Coarse-grid-CFD for the thermal hydraulic investigation of rod-bundles. Proc Appl Math Mech. 15(1):497–498. https://doi.org/10.1002/pamm.201510239
44. Maday Y, Rønquist EM (2002) A reduced-basis element method. J Sci Comput. 17:447–459. https://doi.org/10.1023/A:1015197908587
45. Ohlberger M, Rave S, Schindler F (2017) True error control for the localized reduced basis method for parabolic problems. Model Reduct Parametrized Syst. 169–182
46. Mu L, Zhang G (2019) A domain decomposition model reduction method for linear convection-diffusion equations with random coefficients. SIAM J Sci Comput. 41(3):A1984–A2011. https://doi.org/10.1137/18M1170601
47. Eftang JL, Patera AT (2014) A port-reduced static condensation reduced basis element method for large component-synthesized structures: approximation and a posteriori error estimation. Adv Model Simul Eng Sci. 1(1):3–49. https://doi.org/10.1186/2213-7467-1-3
48. Prusak I, Nonino M, Torlo D, Ballarin F, Rozza G (2023) An optimisation–based domain–decomposition reduced order model for the incompressible navier-stokes equations. Comput Math Appl. 151:172–189. https://doi.org/10.1016/j.camwa.2023.09.039
49. Corigliano A, Dossi M, Mariani S (2013) Domain decomposition and model order reduction methods applied to the simulation of

multi-physics problems in MEMS. Comput Struct. 122:113–127. https://doi.org/10.1016/j.compstruc.2012.12.012

50. Buhr A, Iapichino L, Ohlberger M, Rave S, Schindler F, Smetana K (2020) Localized model reduction for parameterized problems. In handbook on model order reduction. Walter De Gruyter
51. Heinlein A, Klawonn A, Lanser M, Weber J (2021) Combining machine learning and domain decomposition methods for the solution of partial differential equations—a review. GAMM-Mitteilungen 44(1):e202100001. https://doi.org/10.1002/gamm.202100001
52. Quarteroni A, Valli A (1999) Domain decomposition methods for partial differential equations. Oxford University Press
53. Quarteroni A, Quarteroni S (2009) Numerical models for differential problems, vol 2. Springer
54. Diaz AN, Choi Y, Heinkenschloss M (2024) A fast and accurate domain decomposition nonlinear manifold reduced order model. Comput Methods Appl Mech Eng. 425:116943. https://doi.org/10.1016/j.cma.2024.116943
55. Discacciati M, Evans BJ, Giacomini M (2024) An overlapping domain decomposition method for the solution of parametric elliptic problems via proper generalized decomposition. Comput Methods Appl Mech Eng. 418:116484. https://doi.org/10.1016/j.cma.2023.116484
56. Xiao D, Fang F, Heaney CE, Navon IM, Pain CC (2019) A domain decomposition method for the non-intrusive reduced order modelling of fluid flow. Comput Methods Appl Mech Eng. 354:307–330. https://doi.org/10.1016/j.cma.2019.05.039
57. Arcucci R, Quilodrán Casas C, Xiao D, Mottet L, Fang F, Pin W, Pain C, Guo Y-K. A domain decomposition reduced order model with data assimilation (DD-RODA). In 2020) Parallel computing: technology trends. IOS Press, pp 189–198
58. Prusak I (2023) Application of optimisation-based domain–decomposition reduced order models to parameter-dependent fluid dynamics and multiphysics problems. PhD thesis, SISSA
59. Iapichino L, Quarteroni A, Rozza G (2016) Reduced basis method and domain decomposition for elliptic problems in networks and complex parametrized geometries. Comput Math Appl. 71(1):408–430. https://doi.org/10.1016/j.camwa.2015.12.001
60. Wang Z, Li Y (2016) Layer pattern thermal design and optimization for multistream plate-fin heat exchangers—a review. Renewable Sustain Energy Rev. 53:500–514. https://doi.org/10.1016/j.rser.2015.09.003
61. Thompson SK (2012) Sampling, vol 755. John Wiley & Sons
62. Murthy MN (1967). Sampling theory and methods
63. Benner P, Gugercin S, Willcox K (2015) A survey of projection-based model reduction methods for parametric dynamical systems. SIAM Rev. 57(4):483–531. https://doi.org/10.1137/130932715
64. Liu X, Wang Z, Ji H, Gong H (2024 August) Application and comparison of several adaptive sampling algorithms in reduced order modeling. Heliyon 10(15):e34928. https://doi.org/10.1016/j.heliyon.2024
65. Bui-Thanh T, Willcox K, Ghattas O (2008 January) Model reduction for large-scale systems with high-dimensional parametric input space. SIAM J Sci Comput. 30(6):3270–3288. ISSN 1064-8275, 1095-7197. https://doi.org/10.1137/070694855
66. Clenshaw CW, Curtis AR (1960) A method for numerical integration on an automatic computer. Numer Math 2(1):197–205. https://doi.org/10.1007/BF01386223
67. McKay MD, Beckman RJ, Conover WJ (2000) A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. Technometrics 42(1):55–61. https://doi.org/10.1080/00401706.2000.10485979
68. Ju L, Du Q, Gunzburger M (2002) Probabilistic methods for centroidal voronoi tessellations and their parallel implementations. Parallel Comput 28(10):1477–1500. https://doi.org/10.1016/S0167-8191(02)00151-5
69. Eftang JL, Patera AT, Rønquist EM (2010) An “hp” Certified reduced basis method for parametrized elliptic partial differential equations. SIAM J Sci Comput 32(6):3170–3200. https://doi.org/10.1137/090780122
70. Knezevic DJ, Patera AT (2010. ISSN 1064-8275, 1095-7197. January) A Certified reduced basis method for the Fokker–planck equation of dilute polymeric Fluids: FENE dumbbells in extensional flow. SIAM J Sci Comput 32(2):793–817. https://doi.org/10.1137/090759239
71. Buffa A, Maday Y, Patera AT, Prud’homme C, Turinici G (2012) A priori convergence of the greedy algorithm for the parametrized reduced basis method. Esaim: M2an 46(3):595–603. https://doi.org/10.1051/m2an/2011056
72. Maday Y, Stamm B (2013 January) Locally adaptive greedy approximations for anisotropic parameter reduced basis spaces. SIAM J Sci Comput 35(6):A2417–A2441, A2417–A2441. https://doi.org/10.1137/120873868
73. Haasdonk B (2013) Convergence rates of the POD–greedy method. Esaim: M2an 47(3):859–873. https://doi.org/10.1051/m2an/2012045
74. Amsallem D, Zahr MJ, Farhat C (2012 December) Nonlinear model order reduction based on local reduced-order bases. Numer Meth Eng 92(10):891–916. ISSN 0029-5981, 1097-0207. https://doi.org/10.1002/nme.4371
75. Cohen A, Dahmen W, DeVore R, Nichols J (2020 September) Reduced basis greedy selection using random training sets. Esaim: M2an 54(5):1509–1524. ISSN 0764-583X, 1290-3841. https://doi.org/10.1051/m2an/2020004
76. Chellappa S, Feng L, Benner P (2021 December) A training set subsampling strategy for the reduced basis method. J Sci Comput 89(3):63. ISSN 0885-7474, 1573-7691. https://doi.org/10.1007/s10915-021-01665-y
77. Chen W, Hesthaven JS, Junqiang B, Qiu Y, Yang Z, Tihao Y (2018) Greedy nonintrusive reduced order model for fluid dynamics. Aiaa J 56(12):4927–4943. https://doi.org/10.2514/1.J056161
78. Alsayyari F, Perkó Z, Tiberga M, Kloosterman JL, Lathouwers D (2021 January) A fully adaptive nonintrusive reduced-order modelling approach for parametrized time-dependent problems. Comput Methods Appl Mech Eng 373:113483. ISSN 00457825. https://doi.org/10.1016/j.cma.2020.113483
79. Veroy K, Prud’homme C, Rovas D, Patera A (2003) A posteriori error bounds for reduced-basis approximation of parametrized noncoercive and nonlinear elliptic partial differential equations. In 16th AIAA computational fluid dynamics conference, pp 3847
80. Rozza G, Huynh DBP, Patera AT (2008) Reduced basis approximation and a posteriori error estimation for affinely parametrized elliptic coercive partial differential equations: application to transport and continuum mechanics. Arch Computat Methods Eng 15(3):229–275. https://doi.org/10.1007/s11831-008-9019-9
81. Grepl MA, Patera AT (2005) A posteriori error bounds for reduced-basis approximations of parametrized parabolic partial differential equations. Esaim: M2an 39(1):157–181. https://doi.org/10.1051/m2an:2005006
82. Negri F, Manzoni A, Rozza G (2015) Reduced basis approximation of parametrized optimal flow control problems for the stokes equations. Comput Math Appl 69(4):319–336. https://doi.org/10.1016/j.camwa.2014.12.010
83. Veroy K, Patera AT (2005) Certified real-time solution of the parametrized steady incompressible navier–stokes equations: rigorous reduced-basis a posteriori error bounds. Int J Numer Methods Fluids 47(8–9):773–788. https://doi.org/10.1002/fld.867
84. Bond BN, Daniel L (2007) A piecewise-linear moment-matching approach to parameterized model-order reduction for highly nonlinear systems. IEEE Trans Comput-Aided Des Integr Circuits

Syst 26(12):2116–2129. https://doi.org/10.1109/TCAD.2007.907258
85. Fahl M (2000). Trust-region methods for flow control based on reduced order modelling
86. Gong H (2018) Data assimilation with reduced basis and noisy measurement: applications to nuclear reactor cores. PhD thesis, Sorbonne Université
87. Franz T, Zimmermann R, Görtz S, Karcher N (2014 March) Interpolation-based reduced-order modelling for steady transonic flows via manifold learning. Int J Comut Fluid Dyn 28(3–4. 3–4):106–121. https://doi.org/10.1080/10618562.2014.918695
88. Franz T (2015) Reduced-order modeling for steady transonic flows via manifold learning. February https://doi.org/10.24355/dbbs.084-201603210941-0
89. Halder R, Fidkowski KJ, Maki KJ (2021) Local non-intrusive reduced-order modeling using Isomap. AIAA Aviat 2021 Forum
90. Karcher N, Franz T (2022 April) Adaptive sampling strategies for reduced-order modeling. CEAS Aeronaut J 13(2):487–502. ISSN 1869-5582, 1869-5590. https://doi.org/10.1007/s13272-022-00574-6
91. Zhuang Q, Hartmann D, Bungartz H-J, Lorenzi JM (2023 January) Active-learning-based nonintrusive model order reduction. DCE 4(e2). ISSN 2632–6736. https://doi.org/10.1017/dce.2022.39
92. Kapadia H, Feng L, Benner P (2024 February) Active-learning-driven surrogate modeling for efficient simulation of parametric nonlinear systems. Comput Methods Appl Mech Eng 419:116657. ISSN 00457825. https://doi.org/10.1016/j.cma.2023.116657
93. Zhang P, Sheil B, Girolami M (2025 September) Active learning informed proper orthogonal decomposition for reduced order modelling of heat transfer in porous medium. Comput Methods Appl Mech Eng 444:118174. ISSN 00457825. https://doi.org/10.1016/j.cma.2025.118174
94. Rahmati AH, Urban NM, Yoon B-J, Qian X (2025 June). Cost-effective reduced-order modeling via Bayesian active learning
95. Zeng X, Luo G (2017 September) Progressive sampling-based Bayesian optimization for efficient and automatic machine learning model selection. Health Inf Sci Syst 5(11):2. https://doi.org/10.1007/s13755-017-0023-z
96. Turan OT (2020) Reduced order Model base creation with bayesian optimization. PhD thesis, Delft University of Technology
97. Blanchard A, Sapsis T (2021 January) Bayesian optimization with output-weighted optimal sampling. J Comput Phys 425:109901. ISSN 00219991 https://doi.org/10.1016/j.jcp.2020.109901
98. Greif L, Hübschle N, Kimmig A, Kreuzwieser S, Martenne A, Ovtcharova J (2026 March) Structured sampling strategies in Bayesian optimization: evaluation in mathematical and real-world scenarios. J Intell Manuf 37(33):1265–1295. https://doi.org/10.1007/s10845-025-02597-2
99. Benner P, Schilders W, Grivet-Talocia S, Quarteroni A, Rozza G, Miguel Silveira L (2020) Model order reduction: volume 3 applications. De Gruyter
100. Rozza G, Malik H, Demo N, Tezzele M, Girfoglio M, Stabile G, Mola A (2018 November) Advances in reduced order methods for parametric industrial problems in computational fluid dynamics. arXiv:1811.08319 [math]
101. Tezzele M, Demo N, Mola A, Rozza G (2021 February) PyGeM: python geometrical morphing. Softw Impacts 7:100047. ISSN 26659638. https://doi.org/10.1016/j.simpa.2020.100047
102. Gadalla M, Tezzele M, Mola A, Rozza G (2019) BladeX: Python Blade Morphing. JOSS 4(34):1203. https://doi.org/10.21105/joss.01203
103. Salmoiraghi F, Ballarin F, Corsi G, Mola A, Tezzele M, Rozza G (2016) Advances in geometrical parametrization and reduced order models and methods for computational fluid dynamics problems in applied Sciences and engineering: overview and perspectives. In Proceedings of the VII European Congress on Computational Methods in Applied Sciences and Engineering (ECCOMAS Congress 2016), pages Institute of Structural Analysis and Antiseismic Research School of Civil Engineering National Technical University of Athens (NTUA), Crete Island, Greece, Greece. 1013–1031. ISBN 978-618-82844-0-1. https://doi.org/10.7712/100016.1867.8680
104. Ivagnes A, Demo N, Rozza G (2024 April) A shape optimization pipeline for marine propellers by means of reduced order modeling techniques. Numer Meth Eng 125(7):e7426. ISSN 0029-5981, 1097-0207. https://doi.org/10.1002/nme.7426
105. Demo N, Tezzele M, Mola A, Rozza G (2021 February) Hull shape design optimization with parameter space and model reductions, and self-learning mesh morphing. JMSE 9(2):185. https://doi.org/10.3390/jmse9020185
106. Maday Y, Ronquist EM (2004) The reduced basis element method: application to a thermal fin problem. SIAM J Sci Comput 26(1):240–258. https://doi.org/10.1137/S1064827502419932
107. Bressloff NW (2007) Parametric geometry exploration of the human carotid artery bifurcation. J Biomech 40(11):2483–2491. https://doi.org/10.1016/j.jbiomech.2006.11.002
108. Lee S-W, Luca Antiga L, Spence JD, Steinman DA (2008) Geometry of the carotid bifurcation predicts its exposure to disturbed flow. Stroke 39(8):2341–2347. https://doi.org/10.1161/STROKEAHA.107.510644
109. Ding Z, Wang K, Li J, Cong X (2001) Flow field and oscillatory shear stress in a tuning-fork-shaped model of the average human carotid bifurcation. J Biomech 34(12):1555–1562. https://doi.org/10.1016/S0021-9290(01)00148-8
110. Manzoni A, Quarteroni A, Rozza G Model reduction techniques for fast blood flow simulation in parametrized geometries. Numer Methods Biomed Eng, vol. 28, no. 6–7, pp. 604–625, Jun. 2012, https://doi.org/10.1002/cnm.1465
111. Buhmann MD (2000) Radial basis functions. Acta numerica 9:1–38. https://doi.org/10.1017/S0962492900000015
112. Schaback R (2007) A practical guide to radial basis functions. Electron Resource 11:1–12
113. Morris AM, Allen CB, Rendall TCS (2008 November) CFD-based optimization of aerofoils using radial basis functions for domain element parameterization and mesh deformation. Int J Numer Methods Fluids 58(8):827–860. ISSN 0271-2091, 1097-0363. https://doi.org/10.1002/fld.1769
114. Forti D (2012 October) Comparison of shape parametrization techniques for fluid structure interaction problems. PhD thesis, Politecnico di Milano
115. Witteveen J, Bijl H (2009) Explicit mesh deformation using inverse distance weighting interpolation. 19th AIAA Comput Fluid Dyn 3996
116. Witteveen J (2010 January) Explicit and robust inverse distance weighting mesh deformation for CFD. In 48th AIAA Aerospace Sciences Meeting Including the New Horizons Forum and Aerospace Exposition, pp https://doi.org/10.2514/6.2010-165
117. D'Amario A (2014) A reduced-order inverse distance weighting technique for the efficient mesh-motion of deformable interfaces and moving shapes in computational problems. PhD thesis, Politecnico di Milano
118. Ballarin F, D'Amario A, Perotto S, Rozza G (2019 February) A POD-selective inverse distance weighting method for fast parametrized shape morphing. Numer Meth Eng 117(8):860–884. ISSN 0029-5981, 1097-0207. https://doi.org/10.1002/nme.5982
119. Manzoni A, Quarteroni A, Rozza G (2012 October) Shape optimization for viscous flows by reduced basis methods and free-form deformation. Int J Numer Methods Fluids 70(5):646–670. ISSN 0271-2091, 1097-0363. https://doi.org/10.1002/fld.2712

120. Sederberg TW, Parry SR (1986) Free-form deformation of solid geometric models. In Proceedings of the 13th annual conference on Computer graphics and interactive techniques, pp 151–160
121. Garotta F, Demo N, Tezzele M, Carraturo M, Reali A, Rozza G. Reduced order isogeometric analysis approach for pdes in parametrized domains. In 2020) Quantification of uncertainty: improving efficiency and Technology: qUIET selected contributions. Springer, pp 153–170
122. Ballarin F, Manzoni A, Rozza G, Salsa S (2014 September) Shape optimization by free-form deformation: existence results and numerical solution for Stokes flows. J Sci Comput 60(3):537–563. ISSN 0885-7474, 1573-7691. https://doi.org/10.1007/s10915-013-9807-8
123. Rozza G, Manzoni A, Negri F (2012 September) Reduction strategies for pde-constrained optimization problems in haemodynamics. Vienna, Austria
124. Ballarin F (2015). Reduced-order models for patient-specific haemodynamics of coronary artery bypass grafts
125. Padula G, Romor F, Stabile G, Rozza G (2023 August). Generative models for the deformation of industrial shapes with linear geometric constraints: model order and parameter space reductions. arXiv:2308.03662 [math]
126. Siena P, Girfoglio M, Ballarin F, Rozza G (2023) Data-driven reduced order modelling for patient-specific hemodynamics of coronary artery bypass grafts with physical and geometrical parameters. J Sci Comput 94(2):38. https://doi.org/10.1007/s10915-022-02082-5
127. Constantine PG (2015) Active subspaces: emerging ideas for dimension reduction in parameter studies. SIAM
128. Demo N, Tezzele M, Rozza G (2019 November) A non-intrusive approach for the reconstruction of POD modal coefficients through active subspaces. C R Mec 347(11):873–881. https://doi.org/10.1016/j.crme.2019.11.012
129. Tezzele M, Demo N, Stabile G, Mola A, Rozza G (2020 December) Enhancing CFD predictions in shape design problems by model and parameter space reduction. Adv Model Simul Eng Sci 7(1):40. ISSN 2213-7467. https://doi.org/10.1186/s40323-020-00177-y
130. Romor F, Tezzele M, Rozza G (2024 June) A local approach to parameter space reduction for regression and classification tasks. J Sci Comput 99(3):83. ISSN 0885-7474, 1573-7691. https://doi.org/10.1007/s10915-024-02542-0
131. Mola A, Tezzele M, Gadalla M, Valdenazzi F, Grassi D, Padovan R, Rozza G (2019 May) Efficient reduction in shape parameter space dimension for ship propeller blade design. arXiv:1905.09815 [cs
132. Padula G, Romor F, Stabile G, Rozza G (2024 April) Generative models for the deformation of industrial shapes with linear geometric constraints: Model order and parameter space reductions. Comput Methods Appl Mech Eng 423:116823. ISSN 00457825 https://doi.org/10.1016/j.cma.2024.116823.
133. Amsallem D, Cortial J, Farhat C (2010) Towards real-time computational-fluid-dynamics-based aeroelastic computations using a database of reduced-order information. Aiaa J 48(9):2029–2037. https://doi.org/10.2514/1.J050233
134. Washabaugh K, Amsallem D, Zahr M, Farhat C (2012 June) Nonlinear model reduction for CFD problems using local reduced-order bases. In 42nd AIAA Fluid Dynamics Conference and Exhibit, American Institute of Aeronautics and Astronautics. ISBN 978-1-60086-933-4. New Orleans, Louisiana. https://doi.org/10.2514/6.2012-2686
135. Pagani S, Manzoni A, Quarteroni A (2018 October) Numerical approximation of parametrized problems in cardiac electrophysiology by a local reduced basis method. Comput Methods Appl Mech Eng 340:530–558. ISSN 00457825. https://doi.org/10.1016/j.cma.2018.06.003
136. Chasapi M, Antolin P, Buffa A (2023 May) A localized reduced basis approach for unfitted domain methods on parameterized geometries. Comput Methods Appl Mech Eng 410:115997. ISSN 00457825. https://doi.org/10.1016/j.cma.2023.115997
137. Mueller N, Badia S, Zhao Y (2026 April) Reduced basis solvers for unfitted methods on parameterized domains. Comput Methods Appl Mech Eng 451:118610. ISSN 00457825. https://doi.org/10.1016/j.cma.2025.118610
138. Cai W (2017 February) A dimension reduction algorithm preserving both global and local clustering structure. Knowl-Based Syst 118:191–203. ISSN 09507051. https://doi.org/10.1016/j.knosys.2016.11.020
139. Haasdonk B, Dihlmann M, Ohlberger M (2011 August) A training set and multiple bases generation approach for parameterized model reduction based on adaptive grids in parameter space. Math Comput Modell Dyn Syst 17(4):423–442. https://doi.org/10.1080/13873954.2011.547674
140. Zou Z, Kouri D, Aquino W (2019 March) An adaptive local reduced basis method for solving PDEs with uncertain inputs and evaluating risk. Comput Methods Appl Mech Eng 345:302–322. ISSN 00457825. https://doi.org/10.1016/j.cma.2018.10.028
141. Baiges J, Codina R, Idelsohn S (2013) A domain decomposition strategy for reduced order models. application to the incompressible Navier–Stokes equations. Comput Methods Appl Mech Eng 267:23–42. https://doi.org/10.1016/j.cma.2013.08.001
142. Iollo A, Sambataro G, Taddei T (2023) A one-shot overlapping schwarz method for component-based model reduction: application to nonlinear elasticity. Comput Methods Appl Mech Eng 404:115786. https://doi.org/10.1016/j.cma.2022.115786
143. Beattie CA, Borggaard J, Gugercin S, Iliescu T (2006 December) A domain decomposition approach to POD. In Proceedings of the 45th IEEE Conference on Decision and Control, pp 6750–6756. https://doi.org/10.1109/CDC.2006.377642
144. Wang Z, McBee B, Iliescu T (2016 December) Approximate partitioned method of snapshots for POD. J Comput Appl Math 307:374–384. ISSN 03770427. https://doi.org/10.1016/j.cam.2015.11.023
145. Hou TY, Wu X-H (1997) A multiscale finite element method for elliptic problems in composite materials and porous media. J Comput Phys 134(1):169–189. https://doi.org/10.1006/jcph.1997.5682
146. Smetana K, Taddei T (2023) Localized model reduction for nonlinear elliptic partial differential equations: localized training, partition of unity, and adaptive enrichment. SIAM J Sci Comput 45(3):A1300–A1331. https://doi.org/10.1137/22M148402X
147. Henning P, Peterseim D (2013) Oversampling for the multiscale finite element method. Multiscale Model Simul 11(4):1149–1175. https://doi.org/10.1137/120900332
148. Babuška I, Lipton R, Sinz P, Stuebner M (2020) Multiscale-spectral GFEM and optimal oversampling. Comput Methods Appl Mech Eng 364:112960. https://doi.org/10.1016/j.cma.2020.112960
149. Fischer H, Roth J, Wick T, Chamoin L, Fau A (2024 May) More DWR: space-time goal-oriented error control for incremental POD-based ROM for time-averaged goal functionals. J Comput Phys 504:112863. ISSN 00219991. https://doi.org/10.1016/j.jcp.2024.112863
150. Fischer H (2025) Goal-oriented error control for space-time reduced-order modeling using incremental proper orthogonal decomposition. PhD thesis, Universitat Hannover, Hannover
151. Carlberg K (2015) Adaptive h-refinement for reduced-order models. Numer Meth Eng 102(5):1192–1210. https://doi.org/10.1002/nme.4800
152. Chellappa S, Feng L, Benner P (2020 December) Adaptive basis construction and improved error estimation for parametric

nonlinear dynamical systems. Numer Meth Eng 121(23):5320–5349. ISSN 0029-5981, 1097-0207. https://doi.org/10.1002/nme.6462
153. Feng L, Fu G, Wang Z (2021 December) A FOM/ROM hybrid approach for accelerating numerical simulations. J Sci Comput 89(3):61. ISSN 0885-7474, 1573-7691. https://doi.org/10.1007/s10915-021-01668-9
154. Agouzal E, Taddei T (2024 April) Accelerated construction of projection-based reduced-order models via incremental approaches. Adv. Model. Simul. Eng. Sci 11(1):8. ISSN 2213-7467. https://doi.org/10.1186/s40323-024-00263-5
155. Riffaud S (2025 February) Accurate and robust predictions for model order reduction via an adaptive, hybrid FOM/ROM approach. J Comput Phys 523:113677. ISSN 00219991. https://doi.org/10.1016/j.jcp.2024.113677
156. Peherstorfer B, Willcox K (2015 January) Online adaptive model reduction for nonlinear systems via low-Rank updates. SIAM J Sci Comput 37(4):A2123–A2150. ISSN 1064-8275, 1095-7197. https://doi.org/10.1137/140989169
157. Peherstorfer B (2020 January) Model reduction for transport-dominated problems via online adaptive bases and adaptive sampling. SIAM J Sci Comput 42(5):A2803–A2836. ISSN 1064-8275, 1095-7197. https://doi.org/10.1137/19M1257275
158. Terragni F, Valero E, Vega JM (2011 January) Local POD plus Galerkin projection in the unsteady lid-driven cavity problem. SIAM J Sci Comput 33(6):3538–3561. ISSN 1064-8275, 1095-7197. https://doi.org/10.1137/100816006
159. Peherstorfer B, Willcox K (2015 July) Dynamic data-driven reduced-order models. Comput Methods Appl Mech Eng 291:21–41. (ISSN 00457825). https://doi.org/10.1016/j.cma.2015.03.018
160. Etter PA, Carlberg KT (2020 June) Online adaptive basis refinement and compression for reduced-order models via vector-space sieving. Comput Methods Appl Mech Eng 364:112931. ISSN 00457825. https://doi.org/10.1016/j.cma.2020.112931
161. Zhou W, Beerten J (2023) Black Box-based incremental reduced-order modeling framework of inverter-based power systems. IEEE Open J Ind Electron Soc 4:506–518. ISSN 2644–1284. https://doi.org/10.1109/OJIES.2023.3330894
162. Misaka T (2023) Space-time adaptive model order reduction utilizing local low-dimensionality of flow field. J Comput Phys 493:112475. https://doi.org/10.1016/j.jcp.2023.112475
163. Scherding C, Rigas G, Sipp D, Schmid PJ, Sayadi T (2023 November) RONAALP: Reduc-Order Nonlinear Approximation Act Learn Procedure
164. Brand M. Incremental Singular Value Decomposition of Uncertain Data with Missing Values. In Gerhard Goos, Juris Hartmanis, Jan Van Leeuwen, Anders Heyden, Gunnar Sparr, Mads Nielsen, and Peter Johansen (editors) (2002). Computer vision — ECCV 2002, vol 2350. Springer, Berlin Heidelberg, Berlin, Heidelberg, pp 707–720. ISBN 978-3-540-43745-1 978-3-540-47969-7. https://doi.org/10.1007/3-540-47969-4_47
165. Fareed H, Singler JR, Zhang Y, Shen J (2018 March) Incremental proper orthogonal decomposition for PDE simulation data. Comput Math Appl 75(6):1942–1960. https://doi.org/10.1016/j.camwa.2017.09.012
166. Kühl N, Fischer H, Hinze M, Rung T (2024 March) An incremental singular value decomposition approach for large-scale spatially parallel & distributed but temporally serial data – applied to technical flows. Comput Phys Commun 296:109022. ISSN 00104655. https://doi.org/10.1016/j.cpc.2023.109022
167. Di Costanzo E, Kühl N, Marongiu J-C, Rung T (2025) Incremental model order reduction of smoothed-particle hydrodynamic simulations. Int J Numer Methods Fluids 97(12):1571–1594. https://doi.org/10.1002/fld.70012
168. Barrault M, Maday Y, Nguyen NC, Patera AT (2004) An ‘empirical interpolation’ method: application to efficient reduced-basis discretization of partial differential equations. Comptes Rendus. Mathématique 339(9):667–672. https://doi.org/10.1016/j.crma.2004.08.006
169. Chaturantabut S, Sorensen DC (2010) Nonlinear model reduction via discrete empirical interpolation. SIAM J Sci Comput 32(5):2737–2764. https://doi.org/10.1137/090766498
170. Everson R, Sirovich L (1995) Karhunen–loève procedure for gappy data. J Opt Soc Am A 12(8):1657–1664. https://doi.org/10.1364/JOSAA.12.001657
171. Carlberg KT (2011) Model reduction of nonlinear Mechanical systems via optimal projection and tensor approximation. PhD thesis, Stanford University
172. Diaz AN, Choi Y, Heinkenschloss M (2023) Nonlinear-manifold reduced order models with domain decomposition. arXiv preprint arXiv:2312.00713
173. De Pando MF, Schmid PJ, Sipp D (2016 November) Nonlinear model-order reduction for compressible flow solvers using the discrete empirical interpolation method. J Comput Phys 324:194–209. (ISSN 00219991). https://doi.org/10.1016/j.jcp.2016.08.004
174. Dehghan M, Abbaszadeh M (2018 February) A combination of proper orthogonal decomposition–discrete empirical interpolation method (POD–DEIM) and meshless local RBF-DQ approach for prevention of groundwater contamination. Comput Math Appl 75(4):1390–1412. https://doi.org/10.1016/j.camwa.2017.11.012
175. Hoang C, Choi Y, Carlberg K (2021) Domain-decomposition least-squares Petrov–Galerkin (DD-LSPG) nonlinear model reduction. Comput Methods Appl Mech Eng 384:113997. https://doi.org/10.1016/j.cma.2021.113997
176. Ebrahimi M, Yano M (2024 November) A hyperreduced reduced basis element method for reduced-order modeling of component-based nonlinear systems. Comput Methods Appl Mech Eng 431:117254. ISSN 00457825. https://doi.org/10.1016/j.cma.2024.117254
177. De Parga SA, Bravo JR, Hernández JA, Zorrilla R, Rossi R (2023 November) Hyper-reduction for Petrov–Galerkin reduced order models. Comput Methods Appl Mech Eng 416:116298. ISSN 00457825. https://doi.org/10.1016/j.cma.2023.116298
178. Bai F, Wang Y (2020 December) DEIM reduced order model constructed by hybrid snapshot simulation. SN Appl Sci 2(12):2165. ISSN 2523-3971. https://doi.org/10.1007/s42452-020-03958-7
179. Bai F, Wang Y (2022 September) A reduced order modeling method based on GNAT-embedded hybrid snapshot simulation. Math Comput Simul 199:100–132. ISSN 03784754. https://doi.org/10.1016/j.matcom.2022.03.006
180. Bai F, Wang Y (2022 November) DEIM-embedded hybrid snapshot simulation for reduced order model generation. EC 39(10):3321–3353. https://doi.org/10.1108/EC-11-2021-0647
181. Bai F (2026 March) Numerical investigations of reduced-order models with time-varying parameters generated by hybrid training with automatic hyper-reduction in nonlinear diffusion–reaction systems. Math Comput Simul 241:489–526. ISSN 03784754. https://doi.org/10.1016/j.matcom.2025.10.018
182. Farhat C, Grimberg S, Manzoni A, Quarteroni A (2020) Computational bottlenecks for proms: precomputation and hyperreduction. Model Order Reduct 2:181–244
183. Claudio Canuto MYH, Alfio Quarteroni, and Thomas A Zang. 2006) Spectral Methods 285. Springer
184. Stenger F (1974 October) On the convergence and error of the Bubnov-Galerkin method. In Proceedings of the Conference on the Numerical Solution of Ordinary Differential Equations: 19, 20 October 1972. The University of Texas at Austin. Springer, 434–450
185. Bernardi C (1990) A new nonconforming approach to domain decomposition: the mortar element method. Technical report, Universite Pierre at Marie Curie

186. Belgacem FB (1999) The mortar finite element method with Lagrange multipliers. Numerische Math 84(2):173–197. https://doi.org/10.1007/s002110050468
187. Løvgren AE, Maday Y, Rønquist EM (2006) A reduced basis element method for the steady stokes problem. Esaim: M2an 40(3):529–552. https://doi.org/10.1051/m2an:2006021
188. Chen Y, Hesthaven JS, Maday Y (2011) A seamless reduced basis element method for 2D Maxwell's problem: an introduction. In Spectral and High Order Methods for Partial Differential Equations: Selected papers from the ICOSAHOM'09 conference, June 22-26, Trondheim, Norway, Springer, pages 141–152
189. Farhat C, Tezaur R, Toivanen J (2009) A domain decomposition method for discontinuous Galerkin discretizations of Helmholtz problems with plane waves and Lagrange multipliers. Numer Meth Eng 78(13):1513–1531. https://doi.org/10.1002/nme.2534
190. de Castro A, Kuberry P, Tezaur I, Bochev P (2022) A lagrange multiplier partitioned scheme for coupling reduced order models. Technical report, Sandia National Laboratories and Stanford University
191. de Castro AG, Kuberry PA, Tezaur IK, Bochev PB (2023) A partitioned method for reduced ordermodel-finite element model (ROM-FEM) and ROM-ROM couplings with separate reduced bases for interior and interface variables. Technical report, Sandia National Laboratories and Stanford University
192. Iapichino L, Quarteroni A, Rozza G (2012) A reduced basis hybrid method for the coupling of parametrized domains represented by fluidic networks. Comput Methods Appl Mech Eng 221-222:63–82. https://doi.org/10.1016/j.cma.2012.02.005
193. Rozza G (2005) Shape design by optimal flow control and reduced basis techniques: applications to bypass configurations in haemodynamics. PhD thesis, EPFL
194. Rozza G, Veroy K (2007) On the stability of the reduced basis method for Stokes equations in parametrized domains. Comput Methods Appl Mech Eng 196(7):1244–1260. https://doi.org/10.1016/j.cma.2006.09.005
195. Iapichino L (2012) Reduced basis methods for the solution of parametrized PDEs in repetitive and complex networks with application to CFD. PhD thesis, EPFL
196. Martini I, Rozza G, Haasdonk B (2015) Reduced basis approximation and a-posteriori error estimation for the coupled Stokes-Darcy system. Adv Comput Math 41(5):1131–1157. https://doi.org/10.1007/s10444-014-9396-6
197. Eftang JL, Huynh DBP, Knezevic DJ, Ronquist EM, Patera AT (2012) Adaptive port reduction in static condensation. IFAC Proc Volumes 45(2):695–699. https://doi.org/10.3182/20120215-3-AT-3016.00123
198. Huynh DB, Knezevic DJ, Patera AT (2013) A static condensation reduced basis element method: approximation and a posteriori error estimation. Esaim: M2an 47(1):213–251. https://doi.org/10.1051/m2an/2012022
199. Benaceur A, Patera A (2022). Port-reduced reduced-basis component method for steady state Navier–Stokes and passive scalar equations
200. Huynh DBP, Knezevic DJ, Patera AT (2013) A static condensation reduced basis element method: complex problems. Comput Methods Appl Mech Eng 259:197–216. https://doi.org/10.1016/j.cma.2013.02.013
201. Eftang JL, Patera AT (2013) Port reduction in parametrized component static condensation: approximation and a posteriori error estimation. Numer Meth Eng 96(5):269–302. https://doi.org/10.1002/nme.4543
202. Vallaghé S, Patera AT (2014) The static condensation reduced basis element method for a mixed-mean conjugate heat exchanger model. SIAM J Sci Comput 36(3):B294–B320. https://doi.org/10.1137/120887709
203. Kerfriden P, Goury O, Rabczuk T, Bordas SPA (2013) A partitioned model order reduction approach to rationalise computational expenses in nonlinear fracture mechanics. Comput Methods Appl Mech Eng 256:169–188. https://doi.org/10.1016/j.cma.2012.12.004
204. Nitsche J (1971) Über ein variationsprinzip zur lösung von dirichlet-problemen bei verwendung von teilräumen, die keinen randbedingungen unterworfen sind. Abh Math Semin Univ Hamburg 36(1):9–15. ISSN 1865-8784. https://doi.org/10.1007/BF02995904
205. Reed WH, Hill TR (1973) Triangular mesh methods for the neutron transport equation. In Proceedings of the American Nuclear Society
206. Lasaint P, Raviart PA. On a finite element method for solving the neutron transport equation. In 1974) Mathematical aspects of finite elements in partial differential equations. Academic, pp 89–123
207. Cockburn B, Shu C-W (1998 December) The local discontinuous Galerkin method for time-dependent convection-diffusion systems. SIAM J. Numer. Anal 35(6):2440–2463. ISSN 0036-1429, 1095-7170. https://doi.org/10.1137/S0036142997316712
208. Cockburn B (2003. ISSN 0044-2267, 1521-4001. November) Discontinuous Galerkin methods. ZAMM. Z Angew Math Mech 83(11):731–754. https://doi.org/10.1002/zamm.200310088
209. Rivière B (2008) Discontinuous Galerkin methods for solving elliptic and parabolic equations: theory and implementation. SIAM
210. Cockburn B, Karniadakis GE, Shu C-W (2012) Discontinuous Galerkin methods: theory, computation and applications, vol 11. Springer Science & Business Media
211. Demkowicz L, Gopalakrishnan J (2010 April) A class of discontinuous Petrov–Galerkin methods. Part I: the transport equation. Comput Methods Appl Mech Eng 199(23–24):1558–1572. https://doi.org/10.1016/j.cma.2010.01.003
212. Demkowicz L, Gopalakrishnan J (2011) A class of discontinuous Petrov–Galerkin methods. II. Optimal test functions. Numer Methods Partial 27(1):70–105. https://doi.org/10.1002/num.20640
213. Demkowicz L, Gopalakrishnan J, Niemi AH (2012 April) A class of discontinuous Petrov–Galerkin methods. Part III: adaptivity. Appl Numer Math 62(4):396–427. ISSN 01689274. https://doi.org/10.1016/j.apnum.2011.09.002
214. Demkowicz L, Heuer N (2013 January) Robust DPG method for convection-dominated diffusion problems. SIAM J. Numer. Anal 51(5):2514–2537. ISSN 0036-1429, 1095-7170. https://doi.org/10.1137/120862065
215. Demkowicz L, Gopalakrishnan J (2025 July) The discontinuous Petrov–Galerkin method. Acta numerica 34:293–384. ISSN 0962-4929, 1474-0508. https://doi.org/10.1017/S0962492924000102
216. Hesthaven JS, Warburton T (2008) Nodal discontinuous Galerkin methods: algorithms, analysis, and applications. Springer
217. Di Pietro DA, Ern A (2011) Mathematical aspects of discontinuous Galerkin methods, vol 69. Springer Science & Business Media
218. Dolejší V, Feistauer M (2015). Discontinuous Galerkin Method: Analysis and Applications to Compressible Flow. Of Springer series in computational Mathematics. vol 48. Springer, Cham. ISBN 978-3-319-19266-6. https://doi.org/10.1007/978-3-319-19267-3
219. Fidkowski KJ, Chen G (2020) Output-based mesh optimization for hybridized and embedded discontinuous Galerkin methods. Numer Meth Eng 121(5):867–887. https://doi.org/10.1002/nme.6248
220. Cockburn B (2004) Discontinuous Galerkin methods for computational fluid dynamics. Encycl Comput Mech

221. Hesthaven JS, Warburton T (2007) Nodal discontinuous Galerkin methods: algorithms, analysis, and applications. Springer Science & Business Media
222. Benzaken J, Evans JA, Tamstorf R (2024 May) Constructing Nitsche's method for variational problems. Arch Computat Methods Eng 31(4):1867–1896. ISSN 1134-3060, 1886-1784. https://doi.org/10.1007/s11831-023-09953-6
223. Hansbo A, Hansbo P (2002) An unfitted finite element method, based on Nitsche's method, for elliptic interface problems. Comput Methods Appl Mech Eng 191(47–48):5537–5552. https://doi.org/10.1016/S0045-7825(02)00524-8
224. Annavarapu C, Hautefeuille M, Dolbow JE (2012 June) A robust Nitsche's formulation for interface problems. Comput Methods Appl Mech Eng 225-228:44–54. ISSN 00457825. https://doi.org/10.1016/j.cma.2012.03.008
225. Jiang W, Liu Y, Annavarapu C (2022 May) A weighted Nitsche's method for interface problems with higher-order simplex elements. Comput Mech 69(5):1115–1129. ISSN 0178-7675, 1432-0924. https://doi.org/10.1007/s00466-021-02132-z
226. Kamensky D, Hsu M-C, Schillinger D, Evans JA, Aggarwal A, Bazilevs Y, Sacks MS, Hughes TJR (2015 February) An immersogeometric variational framework for fluid–structure interaction: application to bioprosthetic heart valves. Comput Methods Appl Mech Eng 284:1005–1053. ISSN 00457825. https://doi.org/10.1016/j.cma.2014.10.040
227. Burman E, Fernández MA, Frei S (2018 August). A Nitsche-based formulation for fluid-structure interactions with contact
228. Ager C, Schott B, Winter M, Wall WA (2019 July) A Nitsche-based cut finite element method for the coupling of incompressible fluid flow with poroelasticity. Comput Methods Appl Mech Eng 351:253–280. ISSN 00457825. https://doi.org/10.1016/j.cma.2019.03.015
229. Bansal A, Barnafi NA, Pandey DN, Ruiz-Baier R (2025 August). A Nitsche method for Navier–Stokes/generalized poroelasticity interface problems
230. Apostolatos A, Schmidt R, Wüchner R, Bletzinger K-U (2014 February) A Nitsche-type formulation and comparison of the most common domain decomposition methods in isogeometric analysis. Int J Numer Methods Eng 97(7):473–504. https://doi.org/10.1002/nme.4568
231. Elfverson D, Larson MG, Larsson K (2019 June) A new least squares stabilized Nitsche method for cut isogeometric analysis. Comput Methods Appl Mech Eng 349:1–16. ISSN 00457825. https://doi.org/10.1016/j.cma.2019.02.011
232. Ma Y, Wang Z, Yuan X, Chen L (2023) Broadband electromagnetic scattering analysis with isogeometric boundary element method accelerated by frequency-decoupling and model order reduction techniques. The Int Conf Comput Exp Eng Sci 27(2):1–2. https://doi.org/10.32604/icces.2023.09662
233. Pacciarini P, Gervasio P, Quarteroni A (2016) Spectral based discontinuous Galerkin reduced basis element method for parametrized Stokes problems. Comput Math Appl 72(8):1977–1987. https://doi.org/10.1016/j.camwa.2016.01.030
234. Chung SW, Choi Y, Roy P, Moore T, Roy T, Lin TY, Nguyen DT, Hahn C, Duoss EB, Baker SE (2024) Train small, model big: scalable physics simulators via reduced order modeling and domain decomposition. Comput Methods Appl Mech Eng 427:117041. https://doi.org/10.1016/j.cma.2024.117041
235. Romor F, Torlo D, Rozza G (2025) Friedrichs' systems discretized with the DGM: domain decomposable model order reduction and graph neural networks approximating vanishing viscosity solutions. J Comput Phys 531:113915. https://doi.org/10.1016/j.jcp.2025.113915
236. Ferrero A, Iollo A, Larocca F (2018) Global and local POD models for the prediction of compressible flows with DG methods. Numer Meth Eng 116(5):332–357. https://doi.org/10.1002/nme.5927
237. Shah NV, Hess MW, Rozza G (2020 September) Discontinuous Galerkin model order reduction of geometrically parametrized Stokes equation. In Numerical Mathematics and Advanced Applications ENUMATH 2019: European Conference, Egmond aan Zee, vol 30-October 4, pages 551–561. Springer, The Netherlands
238. Li K, Huang T-Z, Liang L, Lanteri S (2023) Simulation of the interaction of light with 3-D metallic nanostructures using a proper orthogonal decomposition-Galerkin reduced-order discontinuous Galerkin time-domain method. Numer Methods Partial 39(2):932–954. https://doi.org/10.1002/num.22911
239. Riffaud S (2020) Reduced-order models: convergence between scientific computing and data for fluid mechanics. PhD thesis, Université de Bordeaux
240. Kaulmann S, Ohlberger M, Haasdonk B (2011) A new local reduced basis discontinuous Galerkin approach for heterogeneous multiscale problems. Comptes Rendus. Mathématique 349(23–24):1233–1238. https://doi.org/10.1016/j.crma.2011.10.024
241. Felix Schindler BH, Ohlberger M, Kaulmann S (2012) The localized reduced basis multiscale method 9:393–403, 09
242. Kaulmann S, Flemisch B, Bernard Haasdonk B, Lie KA, Ohlberger M (2015) The localized reduced basis multiscale method for two-phase flows in porous media. Numer Meth Eng 102(5):1018–1040. https://doi.org/10.1002/nme.4773
243. Ohlberger M, Schindler F (2015) Error control for the localized reduced basis multiscale method with adaptive on-line enrichment. SIAM J Sci Comput 37(6):A2865–A2895. https://doi.org/10.1137/151003660
244. Keil T, Ohlberger M, Schindler F (2023) Adaptive localized reduced basis methods for large scale PDE-constrained optimization. In International Conference on Large-Scale Scientific Computing, Springer, pages 108–116
245. Babuška I, Caloz G, Osborn JE (1994) Special finite element methods for a class of second order elliptic problems with rough coefficients. SIAM J Numer Anal 31(4):945–981. https://doi.org/10.1137/0731051
246. Melenk JM, Babuška I (1996) The partition of unity finite element method: basic theory and applications. Comput Methods Appl Mech Eng 139(1–4):289–314. https://doi.org/10.1016/S0045-7825(96)01087-0
247. Babuška I, Melenk JM (1997) The partition of unity method. Int J Numer Methods Eng 40(4):727–758. https://doi.org/10.1002/(SICI)1097-0207(19970228)40:4%26lt;727::AID-NME86%26gt;3.0.CO;2-N
248. Bordas SPA, Menk A, Natarajan S (2023) Partition of unity methods. John Wiley & Sons
249. Babuska I, Lipton R (2011) Optimal local approximation spaces for generalized finite element methods with application to multiscale problems. Multiscale Model Simul 9(1):373–406. https://doi.org/10.1137/100791051
250. Schleuß J, Smetana K (2022) Optimal local approximation spaces for parabolic problems. Multiscale Model Simul 20(1):551–582. https://doi.org/10.1137/20M1384294
251. Efendiev Y, Hou TY (2009) Multiscale finite element methods: theory and applications, vol 4. Springer Science & Business Media
252. Efendiev Y, Galvis J, Hou TY (2013) Generalized multiscale finite element methods (GMsFEM). J Comput Phys 251:116–135. https://doi.org/10.1016/j.jcp.2013.04.045
253. Chung ET, Efendiev Y, Leung WT (2014) Generalized multiscale finite element methods for wave propagation in heterogeneous media. Multiscale Model Simul 12(4):1691–1721. https://doi.org/10.1137/130926675
254. Efendiev Y, Ginting V, Hou TY (2004) Multiscale finite element methods for nonlinear problems and their applications. Commun

Math Sci 2(4):553–589. https://doi.org/10.4310/CMS.2004.v2.n4.a2

255. Efendiev Y, Hou T (2007) Yalchin Efendiev and T23224321112 Hou. Multiscale finite element methods for porous media flows and their applications. Appl Numer Math 57(5–7):577–596. https://doi.org/10.1016/j.apnum.2006.07.009
256. Efendiev Y, Galvis J, Li G, Presho M (2014) Generalized multiscale finite element methods: oversampling strategies. Int J Mult Comp Eng 12(6):465–484. https://doi.org/10.1615/IntJMultCompEng.2014007646
257. Henning P, Målqvist A (2014) Localized orthogonal decomposition techniques for boundary value problems. SIAM J Sci Comput 36(4):A1609–A1634. https://doi.org/10.1137/130933198
258. Henning P, Målqvist A, Peterseim D (2014) A localized orthogonal decomposition method for semi-linear elliptic problems. Esaim: M2an 48(5):1331–1349. https://doi.org/10.1051/m2an/2013141
259. Chung E, Efendiev Y, Hou TY (2016) Adaptive multiscale model reduction with generalized multiscale finite element methods. J Comput Phys 320:69–95. https://doi.org/10.1016/j.jcp.2016.04.054
260. Gunzburger MD, Peterson JS, Kwon H (1999) An optimization based domain decomposition method for partial differential equations. Comput Math Appl 37(10):77–93. https://doi.org/10.1016/S0898-1221(99)00127-3
261. Taddei T, Xuejun X, Zhang L (2024) A non-overlapping optimization-based domain decomposition approach to component-based model reduction of incompressible flows. J Comput Phys 509:113038. https://doi.org/10.1016/j.jcp.2024.113038
262. Carlberg K, Barone M, Antil H (2017) Galerkin v. least-squares Petrov–Galerkin projection in nonlinear model reduction. J Comput Phys 330:693–734. https://doi.org/10.1016/j.jcp.2016.10.033
263. Keyes DE, Widlund OB (2007) Domain decomposition methods in science and engineering XVI. Springer
264. Nonino M (2020) On the application of the reduced basis method to fluid-structure interaction problems. PhD Thesis, SISSA, Trieste, Italy
265. Liberge E, Hamdouni A (2010 February) Reduced order modelling method via proper orthogonal decomposition (POD) for flow around an oscillating cylinder. J Fluids Struct 26(2):292–311. ISSN 0889-9746. https://doi.org/10.1016/j.jfluidstructs.2009.10.006. Publisher: Elsevier BV
266. Ballarin F, Rozza G (2016 December) POD–Galerkin monolithic reduced order models for parametrized fluid-structure interaction problems. Int J Numer Methods Fluids 82(12):1010–1034. ISSN 0271-2091, 1097-0363. https://doi.org/10.1002/fld.4252. Publisher: Wiley
267. Lieu T, Farhat C, Lesoinne M. Reduced-order fluid/structure modeling of a complete aircraft configuration. 2006. ISSN 0045-7825. August) Reduced-order fluid/structure modeling of a complete aircraft configuration. Comput Methods Appl Mech Eng 195(41–43):5730–5742. https://doi.org/10.1016/j.cma.2005.08.026. Publisher: Elsevier BV
268. Colciago CM, Deparis S, Quarteroni A (2014 August) Comparisons between reduced order models and full 3D models for fluid–structure interaction problems in haemodynamics. J Comput Appl Math 265:120–138. ISSN 0377-0427. https://doi.org/10.1016/j.cam.2013.09.049. Publisher: Elsevier BV
269. Nonino &, Ballarin &, Rozza &, Maday & (2023) A reduced basis method by means of transport maps for a fluid–structure interaction problem with slowly decaying Kolmogorov \$ n \$-width. Adv Comput Sciamp; Eng (ACSE) 1(1):36–58. https://doi.org/10.3934/acse.2023002
270. Hale JK, Koçak H (2012) Dynamics and bifurcations, vol 3. Springer Science & Business Media
271. Khamlich M, Pichi F, Rozza G (2022 October) Model order reduction for bifurcating phenomena in fluid-structure interaction problems. Int J Numer Methods Fluids 94(10):1611–1640. ISSN 0271-2091, 1097-0363. https://doi.org/10.1002/fld.5118
272. Pitton G, Quaini A, Rozza G (2017) Computational reduction strategies for the detection of steady bifurcations in incompressible fluid-dynamics: applications to coanda effect in cardiology. J Comput Phys 344:534–557. https://doi.org/10.1016/j.jcp.2017.05.010
273. Pichi F (2019) Reduced order models for parametric bifurcation problems in nonlinear PDEs. PhD thesis, SISSA, Trieste, Italy
274. Pichi F, Quaini A, Rozza G (2020 January) A reduced order modeling technique to study bifurcating phenomena: application to the gross--pitaevskii equation. SIAM J Sci Comput 42(5):B1115–B1135. ISSN 1064-8275, 1095-7197. https://doi.org/10.1137/20M1313106
275. Pichi F, Strazzullo M, Ballarin F, Rozza G (2022 July) Driving bifurcating parametrized nonlinear PDEs by optimal control strategies: application to Navier–Stokes equations with model order reduction. Esaim: M2an 56(4):1361–1400. ISSN 2822-7840, 2804-7214. https://doi.org/10.1051/m2an/2022044
276. Forti D, Rozza G (2014 March) Efficient geometrical parametrisation techniques of interfaces for reduced-order modelling: application to fluid–structure interaction coupling problems. Int J Comut Fluid Dyn 28(3–4):158–169, 158–169. https://doi.org/10.1080/10618562.2014.932352
277. Wang K, Rallu A, Gerbeau J-F, Farhat C (2011) Algorithms for interface treatment and load computation in embedded boundary methods for fluid and fluid–structure interaction problems. Int J Numer Methods Fluids 67(9):1175–1206. https://doi.org/10.1002/fld.2556
278. Karatzas EN, Stabile G, Atallah N, Scovazzi G, Rozza G (2018 December). A reduced order approach for the embedded shifted boundary FEM and a heat exchange system on parametrized geometries
279. Main A, Scovazzi G (2018) The shifted boundary method for embedded domain computations. Part I: Poisson and Stokes problems. J Comput Phys 372:972–995. ISSN 00219991. https://doi.org/10.1016/j.jcp.2017.10.026. https://linkinghub.elsevier.com/retrieve/pii/S0021999117307799
280. Main A, Scovazzi G (2018) The shifted boundary method for embedded domain computations. Part II: linear advection–diffusion and incompressible Navier–Stokes equations. J Comput Phys 372:996–1026. ISSN 00219991. https://doi.org/10.1016/j.jcp.2018.01.023. https://linkinghub.elsevier.com/retrieve/pii/S0021999118300330
281. Karatzas EN, Ballarin F, Rozza G (2020) Projection-based reduced order models for a cut finite element method in parametrized domains. Comput Math Appl 79(33):833–851. https://doi.org/10.1016/j.camwa.2019.08.003. https://linkinghub.elsevier.com/retrieve/pii/S0898122119303931
282. Karatzas EN, Stabile G, Nouveau L, Scovazzi G, Rozza G (2019) A reduced basis approach for PDEs on parametrized geometries based on the shifted boundary finite element method and application to a Stokes flow. Comput Methods Appl Mech Eng 347:568–587. ISSN 00457825. https://doi.org/10.1016/j.cma.2018.12.040. https://linkinghub.elsevier.com/retrieve/pii/S0045782518306479
283. Balajewicz M, Farhat C (2014) Reduction of nonlinear embedded boundary models for problems with evolving interfaces. J Comput Phys 274:489–504. ISSN 00219991. https://doi.org/10.1016/j.jcp.2014.06.038. https://linkinghub.elsevier.com/retrieve/pii/S0021999114004458
284. Karatzas EN, Stabile G, Nouveau L, Scovazzi G, Rozza G (2020) A reduced-order shifted boundary method for parametrized incompressible Navier–Stokes equations. Comput Methods Appl

Mech Eng 370113273:c. ISSN 00457825. https://doi.org/10.1016/j.cma.2020.113273. http://arxiv.org/abs/1907.10549

285. Zeng X, Stabile G, Karatzas EN, Scovazzi G, Rozza G (2022) Embedded domain reduced basis models for the shallow water hyperbolic equations with the shifted boundary method. Comput Methods Appl Mech Eng 398:115143. ISSN 00457825. https://doi.org/10.1016/j.cma.2022.115143. https://linkinghub.elsevier.com/retrieve/pii/S0045782522003164
286. Karatzas EN, Rozza G A reduced order model for a stable embedded boundary parametrized cahn-hilliard phase-field system based on cut finite elements. URL http://arxiv.org/abs/2009.01596
287. Hansbo A, Hansbo P (2002) An unfitted finite element method, based on Nitsche's method, for elliptic interface problems. Comput Methods Appl Mech Eng 191, August 47–48:5537–5552. https://doi.org/10.1016/S0045-7825(02)00524-8
288. Badia S, Verdugo F, Martín AF (2018 July) The aggregated unfitted finite element method for elliptic problems. Comput Methods Appl Mech Eng 336:533–553. ISSN 00457825. https://doi.org/10.1016/j.cma.2018.03.022
289. Manzoni A, Bonomi D, Quarteroni A (2018) Reduced order modeling for cardiac electrophysiology and mechanics: new methodologies, challenges and perspectives. Math Numer Modeling The Cardiovasc System Appl 115–166
290. Cicci &, Fresca &, Pagani &, Manzoni &, Quarteroni & et al. (2022) Projection-based reduced order models for parameterized nonlinear time-dependent problems arising in cardiac mechanics. MINE 5(2):1–38. https://doi.org/10.3934/mine.2023026
291. Corigliano A, Dossi M, Mariani S (2014) Combined domain decomposition and model order reduction methods for the solution of coupled and non-linear problems. In Proceedings of 11th. World Congress on Computational Mechanics (WCCM XI); 5th. European Congress on Computational Mechanics (ECCM V); 6th. European Congress on Computational Fluid Dynamics (ECFD VI), pp 4115–4123
292. Corigliano A, Dossi M, Mariani S (2015) Model order reduction and domain decomposition strategies for the solution of the dynamic elastic–plastic structural problem. Comput Methods Appl Mech Eng 290:127–155. https://doi.org/10.1016/j.cma.2015.02.021
293. Dossi M (2015 January) Combined Model order reduction and domain decomposition strategies for the solution of non-linear and multi-physics structural p. PhD Thesis, Politecnico di Milano, Milan
294. Zappon E, Manzoni A, Quarteroni A (2023 August). A staggered-in-time and non-conforming-in-space numerical framework for realistic cardiac electrophysiology outputs. arXiv:2308.03884 [math]
295. Deparis S, Forti D, Gervasio P, Quarteroni A (2016 December) INTERNODES: an accurate interpolation-based method for coupling the Galerkin solutions of PDEs on subdomains featuring non-conforming interfaces. Comput Fluids 141:22–41. ISSN 00457930. https://doi.org/10.1016/j.compfluid.2016.03.033
296. Gervasio P, Quarteroni A (2018) Analysis of the INTERNODES method for non-conforming discretizations of elliptic equations. Comput Methods Appl Mech Eng 334:138–166. https://doi.org/10.1016/j.cma.2018.02.004
297. Buffoni M, Telib H, Iollo A (2009) Iterative methods for model reduction by domain decomposition. Comput Fluids 38(6):1160–1167. https://doi.org/10.1016/j.compfluid.2008.11.008
298. Song J, Rui H (2024) A reduced-order Schwarz domain decomposition method based on POD for the convection-diffusion equation. Comput Math Appl 160:60–69. https://doi.org/10.1016/j.camwa.2024.02.016
299. Gander MJ, Kwok F (2012) Best Robin parameters for optimized Schwarz methods at cross points. SIAM J Sci Comput 34(4):A1849–A1879. https://doi.org/10.1137/110837218
300. Maier I, Haasdonk B (2011) An iterative domain decomposition procedure for the reduced-basis-method. Master's thesis, University of Stuttgart
301. Maier I, Haasdonk B (2014) A Dirichlet–Neumann reduced basis method for homogeneous domain decomposition problems. Appl Numer Math 78:31–48. https://doi.org/10.1016/j.apnum.2013.12.001
302. Reyes R, Codina R (2020) Element boundary terms in reduced order models for flow problems: domain decomposition and adaptive coarse mesh hyper-reduction. Comput Methods Appl Mech Eng 368:113159. https://doi.org/10.1016/j.cma.2020.113159
303. Tezaur I, Mota A, Shimizu Y, Barnett J (2022) The Schwarz alternating method for ROM-FOM and ROM-ROM coupling. Technical report, Sandia National Lab.(SNL-CA), Livermore, CA (United States); Stanford University
304. Mota A, Tezaur I, Hoy J (2021) The schwarz alternating method for multi-scale coupling and contact in solid mechanics. Technical report, Sandia National Laboratories and University of Southern California
305. Mota A, Koliesnikova D, Tezaur I, Hoy J (2023) Fundamentally new coupled approach to contact mechanics via the Dirichlet-Neumann Schwarz alternating method. arXiv preprint arXiv:2311.05643
306. Koliesnikova D, Mota A, Tezaur IK (2023) The Dirichlet-Neumann Schwarz alternating method for contact problems in elastodynamics: techniques for reducing artificial oscillations. Technical report, Sandia National Lab.(SNL-CA), Livermore, CA (United States)
307. Vierendeels J, Lanoye L, Degroote J, Verdonck P (2007 June). Implicit coupling of partitioned fluid-structure interaction problems with reduced order models. Computers & structures, vol 85(11-14):Elsevier BV, Publisher, pp 970–976. ISSN 0045-7949. https://doi.org/10.1016/j.compstruc.2006.11.006
308. Aletti M, Lombardi D (2017) A reduced-order representation of the poincaré–Steklov operator: an application to coupled multi-physics problems. Numer Meth Eng 111(6):581–600. https://doi.org/10.1002/nme.5490
309. Nonino M, Ballarin F, Rozza G, Maday Y (2023 January) Projection based semi-implicit partitioned reduced basis method for fluid-structure interaction problems. J Sci Comput 94(1):1573–7691. ISSN 0885-7474. https://doi.org/10.1007/s10915-022-02049-6. Publisher: Springer Science and Business Media LLC
310. Ballarin F, Lee S, Yi S-Y (2024 February) Projection-based reduced order modeling of an iterative scheme for linear thermo-poroelasticity. Results Appl Math 21:100430. ISSN 25900374 https://doi.org/10.1016/j.rinam.2023.100430
311. Riva S, Introini C, Cammi A (2024 November) Multi-physics model bias correction with data-driven reduced order techniques: application to nuclear case studies. Appl Math Modell 135:243–268. ISSN 0307904X. https://doi.org/10.1016/j.apm.2024.06.040
312. Williams CKI, Rasmussen CE (2006) Gaussian processes for machine learning, vol 2. MIT press Cambridge, MA
313. Nielsen MA (2015) Neural networks and deep learning, vol 25. Determination press San Francisco, CA, USA
314. Bonaccorso G (2018) Machine learning algorithms: popular algorithms for data science and machine learning. Packt Publishing Ltd
315. Raissi M, Perdikaris P, Karniadakis GE (2017) Physics informed deep learning (part i): data-driven solutions of nonlinear partial differential equations. arXiv preprint arXiv:1711.10561
316. Raissi M, Perdikaris P, Karniadakis GE (2019) Physics-informed neural networks: a deep learning framework for solving forward and inverse problems involving nonlinear partial differential

equations. J Comput Phys 378:686–707. https://doi.org/10.1016/j.jcp.2018.10.045

317. Trahan C, Loveland M, Dent S (2024) Quantum physics-informed neural networks. Entropy 26(8):649. ISSN 1099-4300. https://doi.org/10.3390/e26080649 https://www.mdpi.com/1099-4300/26/8/649
318. Jagtap AD, Kharazmi E, Karniadakis GE (2020) Conservative physics-informed neural networks on discrete domains for conservation laws: applications to forward and inverse problems. Comput Methods Appl Mech Eng 365:113028. https://doi.org/10.1016/j.cma.2020.113028
319. Cuomo S, Di Cola VS, Giampaolo F, Rozza G, Raissi M, Piccialli F (2022) Scientific machine learning through physics–informed neural networks: where we are and what's next. J Sci Comput 92(3):88. https://doi.org/10.1007/s10915-022-01939-z
320. Zhao C, Zhang F, Lou W, Wang X, Yang J (2024) A comprehensive review of advances in physics-informed neural networks and their applications in complex fluid dynamics. Phys Fluids 36(10). https://doi.org/10.1063/5.0226562
321. Griewank A, Walther A (2008) Evaluating derivatives: principles and techniques of algorithmic differentiation. SIAM
322. Dwivedi V, Parashar N, Srinivasan B (2021) Distributed learning machines for solving forward and inverse problems in partial differential equations. Neurocomputing 420:299–316. https://doi.org/10.1016/j.neucom.2020.09.006
323. Kharazmi E, Zhang Z, Karniadakis GEM (2021) Hp-vpinns: variational physics-informed neural networks with domain decomposition. Comput Methods Appl Mech Eng 374:113547. https://doi.org/10.1016/j.cma.2020.113547
324. Moseley B, Markham A, Nissen-Meyer T (2023) Finite basis physics-informed neural networks (fbpinns): a scalable domain decomposition approach for solving differential equations. Adv Comput Math 49(4):62. https://doi.org/10.1007/s10444-023-10065-9
325. Chen S, Ding Z, Qin L, Wright SJ (2021) A reduced order Schwarz method for nonlinear multiscale elliptic equations based on two-layer neural networks. arXiv preprint arXiv:2111.02280
326. Discacciati N, Hesthaven JS (2024) Model reduction of coupled systems based on non-intrusive approximations of the boundary response maps. Comput Methods Appl Mech Eng 420:116770. https://doi.org/10.1016/j.cma.2024.116770
327. Ruan S (2026) Intrusive and non-intrusive reduced order modeling for flow analysis of typical rod bundle geometries in liquid metal cooled reactors. PhD thesis, Karlsruher Institut für Technologie (KIT) / Karlsruher Institut für Technologie (KIT
328. Xiao D, Yang P, Fang F, Xiang J, Pain CC, Navon IM (2016 May). Non-intrusive reduced order modelling of fluid-structure interactions. Computer methods in applied mechanics and engineering, vol 303. Elsevier BV, Publisher, pp 35–54. ISSN 0045-7825. https://doi.org/10.1016/j.cma.2015.12.029
329. Gerdroodbary MB, Salavatidezfouli S (2025 May). A predictive surrogate model based on linear and nonlinear solution manifold reduction in cardiovascular FSI: A comparative study. Computers in biology and medicine, Elsevier BV, Publisher. 189:109959. ISSN 0010-4825. https://doi.org/10.1016/j.compbiomed.2025.109959
330. Zhang X, Ji T, Xie F, Zheng C, Zheng Y (2022). Publisher: AIP Publishing. May) Data-driven nonlinear reduced-order modeling of unsteady fluid–structure interactions. Phys Fluids 34(5). ISSN 1070-6631, 1089-7666. https://doi.org/10.1063/5.0090394
331. Han R, Wang Y, Qian W, Wang W, Zhang M, Chen G (2022 July) Deep neural network based reduced-order model for fluid–structure interaction system. Phys Fluids 34(7):1089–7666. ISSN 1070-6631. https://doi.org/10.1063/5.0096432. Publisher: AIP Publishing
332. Gobat G, Fresca S, Manzoni A, Frangi A (2023 March) Reduced order modeling of nonlinear vibrating multiphysics microstructures with deep learning-based approaches. Sensors 23(6):3001. ISSN 1424-8220. https://doi.org/10.3390/s23063001. Publisher: MDPI AG
333. Xiao C, Leeuwenburgh O, Lin HX, Heemink A (2019) Non-intrusive subdomain POD-TPWL for reservoir history matching. Comput Geosci 23(3):537–565. https://doi.org/10.1007/s10596-018-9803-z
334. Xiao D, Fang F, Heaney CE, Navon IM, Pain CC (2019) A domain decomposition method for the non-intrusive reduced order modelling of fluid flow. Comput Methods Appl Mech Eng 354:307–330. https://doi.org/10.1016/j.cma.2019.05.039
335. Xiao D, Heaney CE, Fang F, Mottet L, Hu R, Bistrian DA, Aristodemou E, Navon IM, Pain CC (2019) A domain decomposition non-intrusive reduced order model for turbulent flows. Comput Fluids 182:15–27. https://doi.org/10.1016/j.compfluid.2019.02.012
336. Xiao C, Leeuwenburgh O, Lin HX, Heemink A (2019) Subdomain POD-TPWL with local parameterization for large-scale reservoir history matching problems. arXiv preprint arXiv:1901.08059
337. Xiao C, Leeuwenburgh O, Lin HX, Heemink A (2021) Efficient estimation of space varying parameters in numerical models using non-intrusive subdomain reduced order modeling. J Comput Phys 424:109867. https://doi.org/10.1016/j.jcp.2020.109867
338. He J, Durlofsky LJ (2015) Constraint reduction procedures for reduced-order subsurface flow models based on pod–tpwl. Numer Meth Eng 103(1):1–30. https://doi.org/10.1002/nme.4874
339. Arcucci R, Mottet L, Quilodrán Casas CA, Guitton F, Pain C, Guo Y-K (2020). Adaptive domain decomposition for effective data assimilation. In 2019, Revised Selected Papers Euro-par 2019: parallel processing workshops: euro-par 2019 International workshops, vol 25. Springer, Göttingen, Germany, August 26–30, pp 583–595
340. Iyengar N, Rajaram D, Decker K, Perron C, Mavris DN (2022) Nonlinear reduced order modeling using domain decomposition. AIAA SciTech 2022 Forum 1250
341. Iyengar N, Rajaram D, Mavris D (2024) Domain decomposition strategy for combining nonlinear and linear reduced-order models. Aiaa J 62(4):1375–1389. https://doi.org/10.2514/1.J063361
342. Jagtap AD, Karniadakis GE (2020) Extended physics-informed neural networks (xpinns): a generalized space-time domain decomposition based deep learning framework for nonlinear partial differential equations. CiCp 28(5):2002–2041. https://doi.org/10.4208/cicp.OA-2020-0164
343. Hu Z, Jagtap AD, Karniadakis GE, Kawaguchi K (2022) When do extended physics-informed neural networks (xpinns) improve generalization? SIAM J Sci Comput 44(5):A3158–A3182. https://doi.org/10.1137/21M1447039
344. Shukla K, Jagtap AD, Karniadakis GE (2021) Parallel physics-informed neural networks via domain decomposition. J Comput Phys 447(110683):110683. https://doi.org/10.1016/j.jcp.2021.110683
345. Pan X, Xiao D (2024) Domain decomposition for physics-data combined neural network based parametric reduced order modelling. J Comput Phys 519:113452. https://doi.org/10.1016/j.jcp.2024.113452
346. Li K, Tang K, Wu T, Liao Q (2020) D3m: a deep domain decomposition method for partial differential equations. IEEE Access 8:5283–5294. https://doi.org/10.1109/ACCESS.2019.2957200
347. Wuyang L, Xiang X, Yingxiang X (2020) Deep domain decomposition method: elliptic problems. Math Sci Mach Learn 269–286. PMLR
348. Dolean V, Heinlein A, Mishra S, Moseley B (2022) Finite basis physics-informed neural networks as a schwarz domain

decomposition method. In International Conference on Domain Decomposition Methods, Springer, pages 165–172

349. Hrebenshchykova D (2023). Multilevel and distributed physics-informed neural networks for the helmholtz equation
350. Dolean V, Heinlein A, Mishra S, Moseley B (2024) Multilevel domain decomposition-based architectures for physics-informed neural networks. Comput Methods Appl Mech Eng 429:117116. https://doi.org/10.1016/j.cma.2024.117116
351. Snyder W, Tezaur I, Wentland C (2023) Domain decomposition-based coupling of physics-informed neural networks via the schwarz alternating method. arXiv preprint arXiv:2311.00224
352. Feeney A, Zitong L, Bostanabad R, Chandramowlishwaran A (2023) Breaking boundaries: distributed domain decomposition with scalable physics-informed neural pde solvers. In Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis, pp 1–15
353. Wang H, Planas R, Chandramowlishwaran A, Bostanabad R (2022) Mosaic flows: a transferable deep learning framework for solving PDEs on unseen domains. Comput Methods Appl Mech Eng 389:114424. https://doi.org/10.1016/j.cma.2021.114424
354. Lee S, Jang K, Lee S, Cho H, Shin S. (2024 February) Parametric model order reduction by machine learning for fluid-structure interaction analysis. Engineering with Computers, vol 40(1):Springer Science and Business Media LLC, Publisher, pp 45–60. ISSN 0177-0667, 1435-5663. https://doi.org/10.1007/s00366-023-01782-2
355. Farea A, Khan S, Daryani R, Ersan EC, Celebi MS (2025 June). Learning fluid-structure interaction dynamics with physics-informed neural networks and immersed boundary methods. arXiv:2505.18565 [cs]
356. Zhiyong W, Wang H, Chang H, Zhang B, Tao X, Chen Q (2023 November) The Application of Physics-Informed Machine Learning in Multiphysics Modeling in Chemical Engineering. Industrial & engineering chemistry research, vol 62(44):American Chemical Society (ACS), Publisher, pp 18178–18204. ISSN 0888-5885, 1520-5045. https://doi.org/10.1021/acs.iecr.3c02383
357. Ryu Y, Shin S, Lee WB, Jonggeol N (2024 October) Multiphysics generalization in a polymerization reactor using physics-informed neural networks. Chemical engineering Science, vol 298:120385, Elsevier BV, Publisher, pp ISSN 0009–2509. https://doi.org/10.1016/j.ces.2024.120385
358. Yaoyao M, Xiaoyu X, Yan S, Ren Z (2022 February). A Preliminary Study on the Resolution of Electro-Thermal Multi-Physics Coupling Problem Using Physics-Informed Neural Network (PINN). Algorithms, vol 15(2):MDPI AG, Publisher, p 53. ISSN 1999-4893. https://doi.org/10.3390/a15020053
359. Croft TLD (2015) Proper generalised decompositions: theory and applications. PhD thesis, Cardiff University
360. Mienye ID, Swart TG, Obaido G (2024) Recurrent neural networks: a comprehensive review of architectures, variants, and applications. Information 15(9):517. https://doi.org/10.3390/info15090517
361. Li R, Gu X, Shen Y, Li K, Li Z, Zhang Z (2022) Smart and rapid design of nanophotonic structures by an adaptive and regularized deep neural network. Nanomaterials 12(8):1372. https://doi.org/10.3390/nano12081372
362. Pichi F, Moya B, Hesthaven JS (2024) A graph convolutional autoencoder approach to model order reduction for parametrized pdes. J Comput Phys 501:112762. https://doi.org/10.1016/j.jcp.2024.112762
363. Chen Y, Lin Y, Sun X, Yuan C, Gao Z (2025) Tensor decomposition-based neural operator with dynamic mode decomposition for parameterized time-dependent problems. J Comput Phys 533:113996. https://doi.org/10.1016/j.jcp.2025.113996
364. Diab W, Al-Kobaisi M (2025) Temporal neural operator for modeling time-dependent physical phenomena. Sci Rep 15(1). arXiv preprint arXiv:2504.20249. https://doi.org/10.1038/s41598-025-16922-5
365. Jensen M (2004) Discontinuous galerkin methods for Friedrichs systems with irregular solutions. PhD thesis, University of Oxford