Intrusive and Non-intrusive Reduced Order Modeling for Flow Analysis of Typical Rod Bundle Geometries in Liquid Metal Cooled Reactors

Zur Erlangung des akademischen Grades eines

DOKTORS DER INGENIEURWISSENSCHAFTEN (Dr.-Ing.)

von der KIT-Fakultät für Maschinenbau des
Karlsruher Instituts für Technologie (KIT)

angenommene

DISSERTATION

von

M.Sc. Shenhui Ruan

Tag der mündlichen Prüfung:	05.05.2026
Hauptreferent:	Prof. Dr.-Ing. (habil.) Andreas G. Class
Korreferenten:	Prof. Dr.-Ing. Xu Cheng
	Prof. Dr. Gianluigi Rozza (MathLab, SISSA, Italy)

Kurzfassung

Kernenergie gilt als kohlenstoffarme Energiequelle, die dazu beitragen kann, den globalen Klimawandel abzumildern und die Versorgungssicherheit zu gewährleisten. Reaktoren der Generation IV wurden vorgeschlagen, um die Nachhaltigkeit, Sicherheit und Wirtschaftlichkeit von Kernkraftwerken zu verbessern. Flüssigmetallgekühlte schnelle Reaktoren (LMFRs) gelten aufgrund ihrer überlegenen neutronischen und thermohydraulischen Eigenschaften als vielversprechende Konzepte der Generation IV.

Für die Auslegung eines Reaktors spielt die thermohydraulische Analyse eine entscheidende Rolle, um einen sicheren und effizienten Betrieb zu gewährleisten. Numerische Simulationen werden häufig eingesetzt, um die komplexen Phänomene in Kernreaktoren zu untersuchen. Computational Fluid Dynamics (CFD) wird zunehmend für detaillierte Strömungs- und Wärmeübertragungsanalysen in Kernreaktoren verwendet. Hochgenaue CFD-Simulationen sind jedoch nach wie vor rechenintensiv und zeitaufwendig, was ihre Anwendung bei umfangreiche Analysen, z. B. Unsicherheitsquantifizierung, Sensitivitätsanalyse und Optimierung, einschränkt.

Reduced Order Modeling (ROM) ist ein aufstrebendes datengetriebenes Framework, das darauf abzielt, den Rechenaufwand erheblich zu reduzieren und dabei eine akzeptable Genauigkeit zu gewährleisten. Es basiert auf einer Reihe bekannter hochgenauer Lösungen (sogenannte Snapshots). Durch Extraktion der wichtigsten Merkmale aus den Snapshots werden ROMs konstruiert, um die Systemantwort für unbekannte Eingabeparameter zu approximieren. Die Beschleunigung durch ROMs macht sie attraktiv für parametrische Studien und Echtzeitsimulationen.

Die Vorgehensweise zur Erstellung von ROMs ist nicht eindeutig. Es kann allgemein in intrusiven und nicht-intrusiven Methoden unterteilt werden. Intrusive ROMs erfordern den Zugriff auf und die Modifikation der hochgenauen partiellen Differentialgleichungen, während nicht-intrusive ROMs rein datengetrieben sind und als „Black Box“ behandelt werden, um die Beziehung zwischen Parametern und Systemantworten zu approximieren.

Ein Überblick über die vorhandene Literatur zeigt, dass sowohl intrusive als auch nicht-intrusive ROM-Techniken erfolgreich auf verschiedene physikalische Probleme angewendet wurden. Ihre Anwendung in der Kerntechnik ist jedoch noch selten.

Zudem werden die meisten bisherigen ROMs als *globale ROMs* entwickelt, d. h. sie basieren auf dem gesamten Rechengebiet. Dies ist für kleine bis mittelgroße Probleme sinnvoll. Für

großskalige Geometrien ist die Erstellung von CFD-Lösungen für das ROM-Training jedoch rechnerisch zu aufwendig. Um dieser Herausforderung zu begegnen, werden Gebietszerlegungsmethoden integriert. Dies führt zu sogenannten *lokalen ROMs*, im Gegensatz zu *globalen ROMs*, die auf dem gesamten Gebiet basieren.

Die Methodik besteht darin, das gesamte Gebiet anhand geometrischer Merkmale in kleinere Teilgebiete zu unterteilen. Die ROMs können auf lokaler Ebene konstruiert und dann gekoppelt werden, um die globale Lösung zu rekonstruieren. In Anbetracht der weiten Verbreitung sich wiederholender Geometrien ist es möglich, Gebiete zu modellieren, die größer sind als die Trainingsgeometrien. Dieser Ansatz kann den Rechenaufwand für die Erstellung von Snapshots für riesige Modelle erheblich reduzieren.

Kurz gesagt betrachtet diese Dissertation die Entwicklung und Anwendung sowohl intrusiver als auch nicht-intrusiver ROMs für die thermohydraulische Analyse von LMFRs. Die Anwendungen sollen die Leistungsfähigkeit von ROMs für großskalige Modelle demonstrieren, die für kerntechnische Szenarien relevant sind. Das *POD-Galerkin*-ROM wird als intrusive Methode entwickelt, während das *POD mit Interpolation (PODI)*-ROM als nicht-intrusive Methode eingesetzt wird. Beide Techniken werden detailliert erläutert und erklärt. Die Einflüsse der ROM-Einstellungen auf die Modellleistung werden strukturiert untersucht. Für lokale ROMs wird PODI ausgewählt, um ROMs auf Teilgebietsebene zu erstellen, und eine Dirichlet-Neumann-Kopplungsstrategie wird angewendet, um die Einzelteile zum globalen System zusammenzusetzen.

Diese Untersuchungen demonstrieren das Potenzial von ROMs für die thermohydraulische Analyse von LMFRs. POD-Galerkin-ROMs können die Strömungsfelder mit wenigen Snapshots genau vorhersagen. PODI-Techniken können mit geeigneten Modellkonfigurationen eine zufriedenstellende Genauigkeit erreichen. Die Implementierung lokaler ROMs zeigt, dass iterative Schemata in der Lage sind, verschiedene lokale Teilprobleme zu koppeln. Außerdem ist es möglich, dies auf die Approximation eines großen Gebiets mit repetitiven Strukturen zu erweitern.

Die Ergebnisse dieser Studie liefern wertvolle Einblicke in die numerische Analyse von Kernreaktorsystemen. ROM birgt großes Potenzial, die Effizienz und Effektivität thermohydraulischer Simulationen zu verbessern und so zum Fortschritt bei der Auslegung von Kernreaktoren und der Sicherheitsquantifizierung beizutragen. Zusätzlich zeigt das lokale ROM-Framework eine vielversprechende zukünftige Richtung auf, ROMs in kleinen Gebieten zu trainieren und sie auf großskalige Probleme anzuwenden.

Abstract

Nuclear energy is regarded as a low-carbon energy source that can help mitigate global warming and ensure energy supply security. The Generation IV nuclear reactors were proposed to improve the sustainability, safety, and economics of nuclear energy. Among the alternative Generation IV designs, liquid metal cooled fast reactors (LMFRs) are considered the most promising due to their superior neutronic and thermal-hydraulic performance.

Thermal-hydraulic analysis plays a crucial role in reactor design, ensuring safe and efficient operation. Numerical simulations are widely used to investigate these complex phenomena. Recently, Computational Fluid Dynamics (CFD) has increasingly been used to model detailed fluid dynamics and heat transfer in nuclear systems. However, high-fidelity CFD simulations are still computationally expensive and time-consuming, which limits their application to many-query simulations, e.g., uncertainty quantification, sensitivity analysis, and optimization.

Reduced order modeling (ROM) is an emerging data-driven framework that aims to reduce computational cost while maintaining acceptable accuracy. Given a set of known high-fidelity solutions (i.e., snapshots), a reduced system is constructed based on the most dominant features extracted from the snapshots. Then, ROMs are employed to approximate the system response for unknown input parameters. In many applications, ROM acceleration is in the order of 1000 or more, making them attractive for parametric studies and real-time simulations.

The framework for constructing ROMs is not unique. They can be generally categorized into intrusive and non-intrusive methods. Intrusive ROMs require access to and modification of the high-fidelity partial differential equations. In contrast, non-intrusive ROMs are purely data-driven and treated as "black boxes" to approximate the relationship between parameters and system responses.

Through reviewing the existing literature, both intrusive and non-intrusive ROM techniques have been successfully applied to various physical problems. However, their applications in nuclear engineering are still limited.

Additionally, most existing ROMs are developed as *global ROMs*, i.e., built on the whole computational domain. That is feasible for small to medium-scale problems. However, for large-scale geometries, preparing CFD solutions for ROM training is computationally prohibitive. To tackle this challenge, domain decomposition methods are integrated. That leads to the so-called *local ROMs*, as opposed to *global ROMs*.

The local methodology consists of several steps. Firstly, the computational domain is divided into smaller subdomains based on its geometrical features. The ROMs can be built for each part and then coupled to reconstruct the global solution. Given the widespread presence of repetitive structures, it can build domains larger than the training geometries. Therefore, the approach can significantly reduce the computational cost for generating snapshots for large scale models.

In short, this dissertation focuses on the development and application of both intrusive and non-intrusive ROMs for fluid dynamics in LMFRs. The applications aim to demonstrate the capabilities of ROMs for large models relevant to nuclear engineering scenarios. The *POD-Galerkin* ROM is selected as an intrusive method, while the *POD with interpolation (PODI)* ROM is employed as a non-intrusive method. Both techniques are explained in detail. The influence of ROM settings on model performance is structurally investigated. For local ROMs, PODI is selected to build subdomain-level ROMs, and a Dirichlet-Neumann coupling strategy is adopted to assemble those partitions to form the global system.

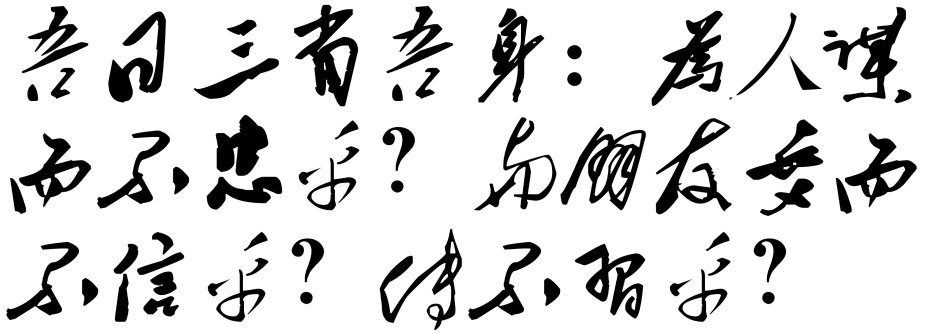
This research demonstrates the capabilities of ROMs for thermal-hydraulic analysis of LMFRs. POD-Galerkin ROMs can accurately predict the flow fields based on a few snapshots. PODI techniques can reach satisfactory accuracy with appropriate model configurations. The implementation of local ROMs shows that iterative schemes can couple multiple local subproblems. It is feasible to approximate a domain that is larger than the one used to generate snapshots.

The outcomes of this study provide valuable insights into the numerical analysis for nuclear reactor systems. ROM has great potential to enhance the efficiency and effectiveness of thermal-hydraulic simulations, advancing nuclear reactor design and safety quantification. Additionally, the local ROM framework demonstrates a promising direction. That denotes training ROMs in small domains while applying them to large-scale problems.

To all I love, and to those who love me.
献给所有我所爱的人，和爱我的人。

A man, 30 years old, risking a glance back.
Two young guys, 18 and 26, stared forward.

They are greeting each other,
across rivers and mountains,
spatially and temporally.



Contents

Acronyms and symbols

Acronyms

1D	One-Dimensional
2D	Two-Dimensional
3D	Three-Dimensional
AE	Autoencoder
AI	Artificial Intelligence
ANN	Artificial Neural Network
AS	Active Subspace
BC	Boundary Condition
CFD	Computational Fluid Dynamics
CG	Continuous Galerkin
CGCFD	Coarse-Grid Computational Fluid Dynamics
CNN	Convolutional Neural Network
CV	Control Volume
DMD	Dynamic Mode Decomposition
DD	Domain Decomposition
DDNN	Data-Driven Neural Network
D-D	Dirichlet-to-Dirichlet Conditions at Interfaces
D-N	Dirichlet-to-Neumann Conditions at Interfaces
DNN	Deep Neural Network
DEIM	Discrete Empirical Interpolation Method
DG	Discontinuous Galerkin
DGRBEM	Discontinuous Galerkin Reduced Basis Element Method

DNS	Direct Numerical Simulation
DOF	Degree of Freedom
FE	Finite Element
FEM	Finite Element Method
FFD	Free-Form Deformation
FOM	Full Order Model
FVM	Finite Volume Method
FSI	Fluid-Structure Interaction
GFEM	Generalized Finite Element Method
HF	High-Fidelity
ITES	Institute for Thermal Energy Technology and Safety
KIT	Karlsruher Institute for Technology
LES	Large Eddy Simulation
LMFRs	Liquid Metal-Cooled Faster Reactors
LS	Least Squares
LSG	Least Squares Galerkin
LSTM	Long Short-Term Memory
ML	Machine Learning
MOR	Model Order Reduction
MSE	Mean Squared Error
MsFEM	Multiscale Finite Element Method
NN	Neural Network
N-N	Neumann-to-Neumann Conditions at Interfaces
NS	Navier-Stokes
ODE	Ordinary Differential Equation
PDE	Partial Differential Equation
PINN	Physics-Informed Neural Network
POD	Proper Orthogonal Decomposition
PODI	Proper Orthogonal Decomposition with Interpolation

PoU	Partition of Unity
PUM	Partition of Unity Method
RANS	Reynolds-Averaged Navier-Stokes
RB	Reduced Basis
RBEM	Reduced Basis Element Method
RBHM	Reduced Basis Hybrid Method
RBM	Reduced Basis Method
RBF	Radial Basis Function
Re	Reynolds Number
RDF	Reduced Basis, Domain Decomposition, and Finite Element Method
ROM	Reduced Order Model
R-R	Robin-to-Robin Conditions at Interfaces
SCRBEM	Static Condensation Reduced Basis Element Method
SGS	Sub-grid-scale
SM	Spectral Method
SQP	Sequential Quadratic Programming
SST	Shear Stress Transport
SVD	Singular Value Decomposition

Latin symbols and variables

A	Truncated POD coefficient matrix or matrix of the divergence term
A_F	Full POD coefficient matrix
b	Bias term in neural network
B	Matrix of the diffusion term
B^L	Matrix of the diffusion term corresponding to the lifting functions
c	Unknowns in an algebraic system or superscript for the coarse mesh
C	Matrix of the convection term
C^L	Matrix of the convection term corresponding to the lifting functions

C^{L1}	Matrix of the convection term corresponding to the lifting functions
D	Rod diameter
D_h	Hydraulic diameter
D_w	Wire diameter
E	Matrix of the turbulent diffusion term
E^L	Matrix of the turbulent diffusion term corresponding to the lifting functions
$\mathcal{E}$	Relative error norm
$\mathcal{E}^2$	Truncated POD energy norm
$\mathcal{E}_{N_{\text{RB}}}$	Truncated POD energy norm with N_{RB} POD modes
$E(\kappa)$	Turbulent eddy's energy spectrum function
f	Source term or subscript for fluid or superscript for the fine mesh
$\mathcal{F}$	Source term matrix or subscript for the Frobenius norm
F_1	Blending function in the SST k-ω model
F_2	Blending function in the SST k-ω model
F^L	Matrix of the source term corresponding to the lifting functions
F_2	Blending function in the SST k-ω model
$\mathbf{g}$	Boundary condition vector
$\mathbf{g}_N$	Neumann boundary condition vector
G	Matrix of the divergence of the convection term in the pressure Poisson equation
$\mathbf{G}$	Snapshot matrix of Neumann conditions
h	Subscript for high-fidelity model
H	Hilbert space
H_w	Wire pitch
i	Subscript for indices
$\mathbf{I}$	Identity matrix
$\mathcal{I}$	Set of indices
j	Subscript for indices
$\mathcal{J}$	Optimization functional
k	Turbulent kinetic energy or Subscript for indices

L	Longitudinal length of the computational domain
m	Subscript for indices
M	Mass matrix or subscript for matrix sizes
$\mathcal{M}$	True solution manifold
$\mathcal{M}_h$	High-fidelity numerical solution manifold
n	Subscript for indices
$\mathbf{n}$	Surface outward normal vector
N	Number of snapshots or dimensionality of POD basis
N_h	Dimensionality of the high-fidelity model
$N_{\mathcal{P}}$	Dimensionality of the parameter space
$N_{\tilde{u}}$	Number of parameters for the online stage
N_μ	Number of parameters
N_Ω	Number of subdomains
$\hat{N}_\Omega$	Number of the reference subdomain
N_Γ	Number of interfaces
$\hat{N}_\Gamma$	Number of the reference interface
N^{RB}	Dimension of the reduced subspace
$N_{\mathbf{u}}^{\mathrm{RB}}$	Dimension of the velocity reduced subspace
N_p^{RB}	Dimension of the pressure reduced subspace
N_{train}	Number of training samples
N_{test}	Number of testing samples
$\mathbf{n}$	Outward unit normal vector
p	Pressure
P	Rod pitch of a bundle
P_k	Stress term in k transport equation of the SST k-ω model
P^n	n-order polynomial function
P_m^n	m-th element of the n-order polynomial function
$\mathcal{P}$	Parameter space
q	Test functions of the pressure field

r	Subscript
$\mathbb{R}$	Set of real numbers
s	Subscript for solid
$\mathbf{S}$	Snapshot matrix
S_t	Blending function in the SST k-ω model
t	Time
T	Geometrical transformation
u	Physical variable
$\mathbf{u}$	Velocity vector
u'	Temporal fluctuation of physical variable
$\dot{u}$	Time dependent variable
u_a	Uniform inflow profile
$\mathbf{u}_D$	Dirichlet boundary condition vector
$\mathbf{u}^L$	Lifting function
u_{in}	Inflow condition
u_{max}	Maximum value of a power law profile
$\mathbf{u}_o$	Homogeneous Dirichlet condition
u_p	Power law inflow profile
U	Velocity magnitude
$\mathcal{U}$	Functional space
v	Basis function
$\mathbf{v}$	Basis function vector
$\mathcal{V}$	Functional space
$\mathcal{V}_N$	Reduced subspace
w	Weighting factor or basis function
$\mathcal{W}$	Functional space
W_1	Linear transformation matrix used in the active subspace method
$\mathbf{x}$	Spatial coordinate vector
x	Spatial coordinate in x-direction or independent variable

y	Spatial coordinate in y-direction or dependent variable
y^+	Dimensionless wall distance
z	Spatial coordinate in z-direction
$\mathcal{Z}$	Regression/surrogate model

Greek symbols and variables

α	POD coefficient
$\boldsymbol{\alpha}$	POD coefficient vector
β	POD coefficient or empirical constant in the SST k-ω model
β^*	Empirical constant in the SST k-ω model
$\boldsymbol{\beta}$	POD coefficient vector
Γ	Boundary of the computational domain
Γ_D	Dirichlet boundary
Γ_N	Neumann boundary
Γ_w	No-slip wall
Γ_{sym}	Symmetric boundary
$\hat{\Gamma}$	Reference boundary
ϵ	Turbulent dissipation rate
$\epsilon_{\mathbf{u}}$	Relative L^2 error norm for velocity
ε	Error norm of ROMs
θ	Switch factor in $0, 1$
κ	Turbulent wave number
λ	Eigenvalue or thermal conductivity or relaxation factor
μ	Parameter
μ'	Unknown parameter
$\boldsymbol{\mu}$	Parameter vector
μ_{new}	New parameter that not used to construct ROMs
μ^L	Factor for the Dirichlet lifting functions

$\boldsymbol{\mu}_D$	Parameter vector for Dirichlet conditions
$\boldsymbol{\mu}_N$	Parameter vector for Neumann conditions
ν	Kinematic viscosity
ν_{eff}	Effective viscosity
ν_t	Turbulent (eddy) viscosity
ρ	Fluid density
σ	Singular value or activation function in neural networks
σ_k	Empirical constant in the k transport equation of the SST k-ω model
σ_w	Empirical constant in the ω transport equation of the SST k-ω model
σ_{w2}	Empirical constant in the ω transport equation of the SST k-ω model
Σ	Covariance of the Gaussian distribution
ς	Coefficient for the linear combination
τ	Shear stress tensor
τ_u	Dirichlet boundary penalty factor
τ_g	Neumann boundary penalty factor
$\boldsymbol{\xi}$	Geometrical parameter vector or POD coefficients for the eddy viscosity
ϕ	Basis function
$\boldsymbol{\phi}$	Basis function vector
φ	Radial basis function
$\boldsymbol{\varphi}$	Basis function vector
Φ	Functional space or truncated POD mode matrix
Φ_F	Full POD mode matrix
Φ^*	POD mode matrix for Dirichlet datasets at interfaces
Ψ	Functional space
Ψ^*	POD mode matrix for Neumann datasets at interfaces
Ω	Computational domain
$\overline{\Omega}$	Closure of the computational domain
$\hat{\Omega}$	Reference domain
$\partial\Omega$	Boundary of the computational domain

ω	Specific rate of turbulent dissipation

Operators and math symbols

$\mathcal{L}$	Linear differential operator
$m(\cdot)$	Mean value
$\mathcal{N}$	Nonlinear differential operator or normal distribution
$\mathcal{Q}$	General differential operator
$^{\mathrm{T}}$	Transpose operator for matrices
$\cdot$	Dot product
$\times$	Cross product
$\|\cdot\|_{L^2}$	L^2 norm
$\cap$	Intersection
$\cup$	Union
Σ	Summation
$(\cdot,\cdot)$	Inner product

Superscript

L	Lifting function
L^2	L^2 space
N	Dimensionality
RB	Reduced basis
T	Matrix transpose
2	Square
-1	Matrix inverse
$*$	Interface-based data
$\hat{\cdot}$	Data defined at generic subdomains
$\tilde{\cdot}$	Test data

Subscript

D	Dirichlet boundary
g	Neumann data
h	High fidelity data
N	Neumann boundary or reduced subspace
p	Pressure field
$\mathcal{P}$	Parameter space
$\mathbf{u}$	Velocity field
in	Inlet boundary
out	Outlet boundary
test	Test data
train	Training data
μ	Parameter
Ω	Domain
Γ	Face
$[mn]$	Interface $[mn]$ of Ω_m and Ω_n
$*$	Unknown data

List of Figures

List of Tables

1 Introduction and motivation

This chapter briefly outlines the background of this doctoral research. Nuclear energy is regarded as an alternative for long-term sustainable development. To ensure the safety of nuclear reactors, numerical techniques have become standard procedures for analyzing their thermal-hydraulic characteristics, especially the latest fourth-generation reactor designs. Nevertheless, the high-fidelity simulation methods are accurate but expensive. Thus, data-driven reduced order models are proposed to achieve efficient and accuracy-preserving parameter computations. Additionally, nuclear cores contain numerous repeating geometrical patterns. Hence, the combination of domain decomposition and reduced order model can be effective in such structures, yet challenging. Hence, given the current state of numerical analysis in nuclear engineering, the motivations for this study are clarified.

1.1 Background

Statistics indicate a continuous increase in global temperature with the growth of CO_2 level since 1960 [1; 2]. It is widely acknowledged that global warming poses unprecedented risks to life [3]. The growing and persistent consumption of fossil fuels is the main contributor to rising CO_2 in the atmosphere. Thus, the search for *clean* alternatives to fossil fuels has become a focus in the energy industry [4].

Nuclear power is considered a *clean electricity source* [5], generating nearly zero CO_2 or other greenhouse gases [6; 7]. It also avoids the air pollutants typically produced by burning fossil fuels [5].

Nuclear fission energy also has several advantages over renewable energy technologies, e.g., solar, wind, and hydropower [8]. Wind and solar are highly dependent on weather conditions. Hydropower can vary significantly due to seasonal variability in river flow. In contrast, nuclear power plants can provide continuous and consistent base-load electricity. These features make nuclear power a promising alternative to traditional fossil fuels.

In 2002, the *Generation IV International Forum* (GIF) reviewed existing designs and proposed next-generation concepts (i.e., Generation IV) to enhance the safety, reliability, and sustainability of nuclear power [9]. Six technologies were selected as candidates to accelerate research and commercialization of next-generation nuclear reactors [9].

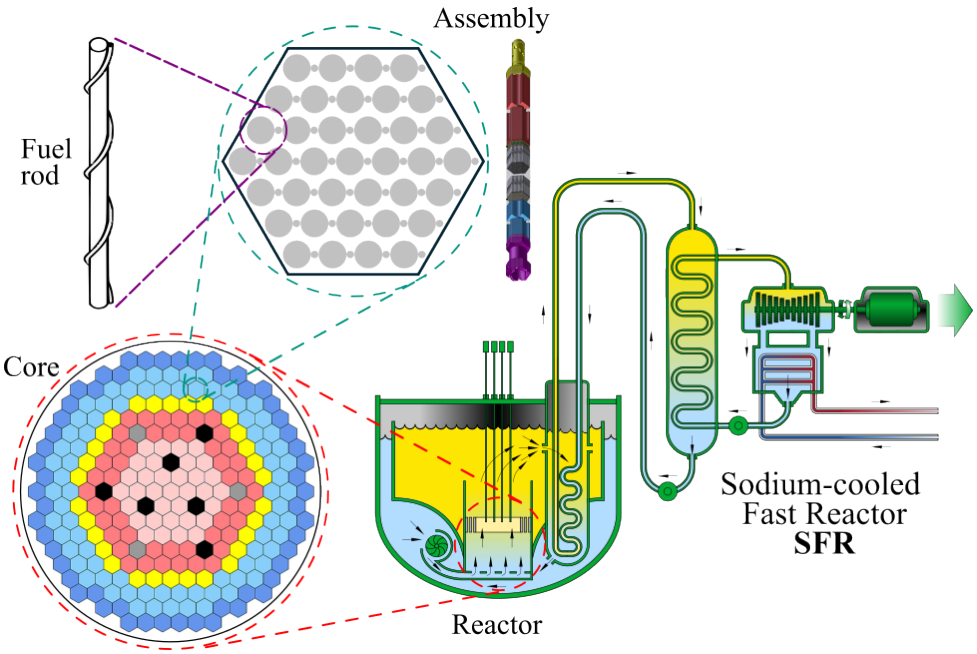


Figure 1.1: Schematic of a typical Sodium-cooled Fast Reactor (SFR), including a core, a fuel assembly, and a fuel rod. Figures taken with permission from [11; 12], copyright owned by Elsevier.

Among the candidate designs, two use liquid metal coolants: the sodium-cooled fast reactor (SFR) and lead-cooled fast reactor (LFR). Both are known as *Liquid Metal-cooled Fast Reactors* (LMFRs). The GIF Technical Roadmap [10] highlights the strong viability of LMFRs for near-term deployment, emphasizing the necessity to investigate their heat transport and fluid dynamic characteristics.

As illustrated in Fig. 1.1, a typical SFR contains four scales of components, which are the reactor, core, fuel assembly, and rod. The most important part of a reactor is the core, where the thermal energy is generated from fission reactions. A core consists of hundreds of fuel assemblies. Each assembly contains multiple fuel rods arranged to form parallel, geometrically symmetric coolant channels. A wire wraps around each rod to provide mechanical support and enhance transverse mixing between the flow channels. A bunch of rods and wires then comprises a *wire-wrapped rod bundle*. The rods without wires can assemble a *bare-rod bundle*. Note that the number of rods in a bundle is generally small compared to an assembly.

The core is considered the most critical component in a nuclear plant. Its flow and energy transport characteristics significantly influence the efficiency and safety of the whole power plant [13; 14].

Decades of experimental and numerical research have been employed to advance the understanding of essential thermal-hydraulic phenomena in nuclear systems. With the rapid growth in computational power, numerical methods have become standard tools for designing and optimizing LMFRs [15].

1.2 Numerical techniques in nuclear engineering

A nuclear power plant exhibits multiple length scales [14]. When designing the core of a reactor, engineers must consider phenomena occurring at these scales, i.e., reactor, core/assembly, and bundle, as illustrated in Fig. 1.2. Their sizes range from several meters to sub-millimeters. Consequently, numerical methods with varying computational costs and spatial resolutions have been developed to simulate different physical scales [15], as shown in Fig. 1.2.

Now, let us start discussing numerical analysis for the three scales [16]. For the entire reactor system, detailed modeling is not feasible. Thus, a simplified model is employed, in which the whole reactor is discretized into a few volumes. Each block, or a few blocks, roughly represents a component, such as the core and steam generator. The so-called *system code* aims to capture the system-level thermal-hydraulic response, while neglecting local processes. Next, the *subchannel approach* utilizes coarse control volumes to represent parallel flow channels in the cores. It can better represent the mass and heat transport inside the core. However, some essential phenomena, such as wall heat transfer, flow resistance, and inter-channel mixing (both mass and heat transfer), are computed using experiment-based (i.e., empirical) correlations, rather than numerical modeling. Higher-resolution thermal-hydraulic behaviour can be resolved using *Computational Fluid Dynamics* (CFD). It focuses on small geometries and adopts a much finer spatial and temporal discretization.

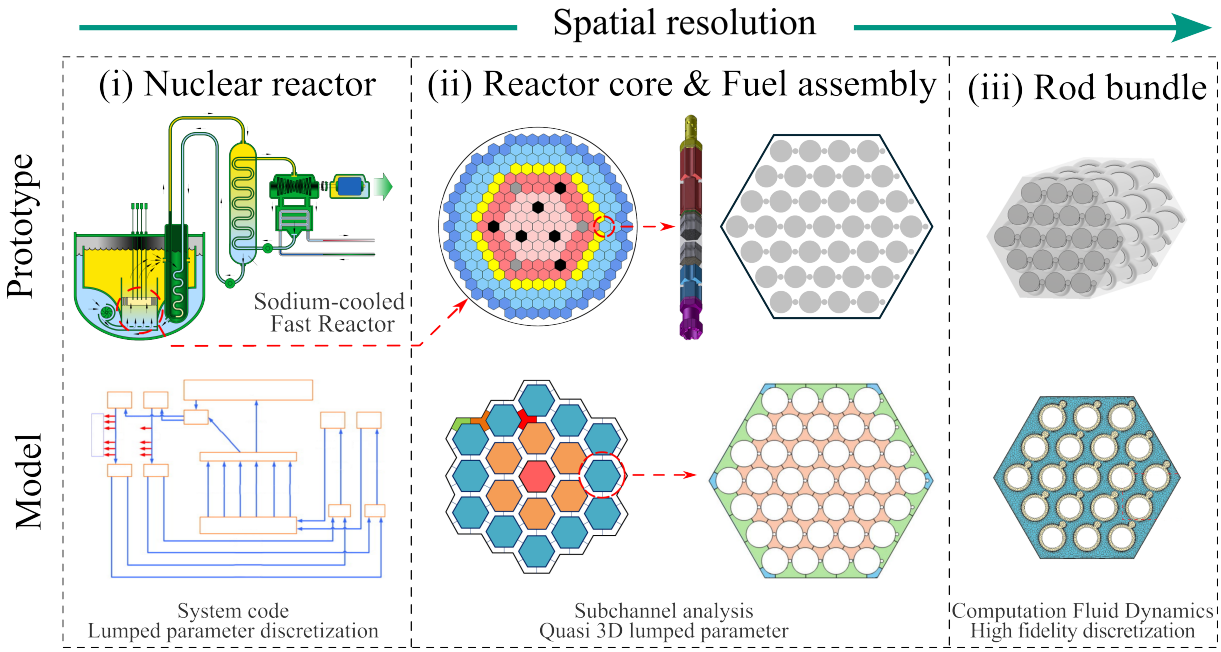


Figure 1.2: Numerical tools utilized for the thermal-hydraulic design of a nuclear reactor: (i) 1D lumped parameter system codes analyze system-level responses. (ii) The quasi-3D subchannel approach approximates flow and heat transport within cores. (iii) High-fidelity CFD, adopted to resolve small-scale features in rod bundles. Figures taken with permission from [12; 17; 18; 11; 19; 20; 21], copyright owned by Elsevier.

Both the one-dimensional (1D) *System* approach and quasi-three-dimensional (3D) *Subchannel* analysis [1] are built in frames of the so-called *lumped parameter* approach. They are based on the assumption that a system (or a component) is divided into homogenized control volumes. In every volume, only a single averaged value of each physical quantity is defined. Spatial gradients of the quantities are neglected. This extremely coarse spatial discretization enables rapid simulations for large models. In addition, simplified governing equations are integrated into the framework.

The techniques are practical when hardware capability is inadequate [13; 14]. Even today, they are generally used to design and analyze nuclear systems. As the two methods are relevant to the topic of this dissertation, a brief description and literature review are presented in Sec. 2.1.

CFD techniques were proposed to improve the modeling of small-scale phenomena. They contain two essential steps. Firstly, the computational domain is represented by a large number of finite discrete points, or elements or cells, collectively referred to as a mesh. The *Partial Differential Equations* (PDEs) are discretized in each element and then assembled into an algebraic system. In general, for transient problems, small time steps are required to well capture rapid changes. To conclude, the CFD methods employ high-resolution discretization in both spatial and temporal spaces, and they are referred to as high-fidelity numerical simulation methods hereafter.

A comparative ranking of the methods based on computational complexity and spatial resolution is shown in Fig. 1.3. It should be noted that, for system and subchannel codes, turbulence contributions are computed through empirical correlations. In contrast, CFD approaches are specifically designed to compute turbulent flow phenomena numerically.

Nevertheless, turbulence involves multiscale features in both space and time [22]. To model or resolve turbulence across different scales, several primary strategies have been developed [23; 24] (see Fig. 1.3). These are: (i) *Reynolds-averaged Navier-Stokes* (RANS); (ii) *Large Eddy Simulation* (LES); (iii) *Detached Eddy Simulation* (DES); (iv) *Direct Numerical Simulation* (DNS). Note that DES is a hybrid technique that dynamically switches between RANS and LES depending on the local mesh resolution. The choice of the model depends on the problems, balancing computational cost and accuracy. Further details, including their applications in nuclear engineering, are discussed in Sec. 2.2.

Despite exponential growth in computational resources over the past decade, both DNS and LES remain prohibitively expensive for most nuclear engineering applications. Consequently, RANS continues to be regarded as the standard approach, both now and for the foreseeable future [26].

[1] Subchannel codes divide the axial flow channels between fuel rods into numerous 1D *subchannels*. The 1D governing equations are used to model axial fluid and heat transport, while transverse crossflow and mixing between neighboring channels are approximated employing empirical formulations. Thus, it's not purely 1D, and the essential 3D effects, i.e., flow and heat interactions, are partially represented. Thus, it is considered to be quasi-3D.

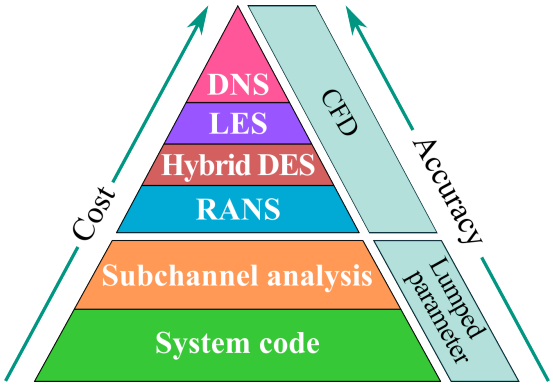


Figure 1.3: Ranking of lumped parameter and CFD methods in terms of computational cost and accuracy [25].

1.3 Computational fluid dynamics and reduced order models

1.3.1 Necessity of parametric simulation

Due to specific fluid properties and flow conditions in Gen-IV reactors, experimental investigations are still fairly limited. Thus, high-fidelity numerical techniques are extensively employed to analyze fluid mechanics and heat transfer in components of nuclear systems (e.g., fuel assemblies and heat exchangers) [27; 28].

For the design of a nuclear system or its components, there are a series of scenarios that should be analyzed [13; 14]: (i) design and optimization of the structures, (ii) design and off-design operation conditions, (iii) accidents quantification, (iv) sensitivity and uncertainty analysis to guarantee the safety margin, etc. That requires conducting numerous parametric simulations, which are challenging even when we incorporate the less expensive RANS model.

It is worth noting that the computational complexity increases significantly for modeling the wire-wrapped bundle [29]. Because the global length of a bundle is several meters, while the transverse width of the narrow gap between neighbouring rods is less than a millimeter. That requires a fine mesh to capture the local behaviours.

1.3.2 Reduced order model for parametric simulation

The limitation of computational resources has prompted the research community to seek more efficient techniques for parametric analyses. Considering the relevance to this dissertation, *Reduced Order Model* (ROM) is regarded as a technique capable of achieving fast and cheap high-resolution approximations for a wide range of parameters, and thus dealing with these difficulties [30].

The ROM belongs to the data-driven framework, which uses data to build highly efficient numerical models. It is constructed from the postprocessing of a set of precomputed high-fidelity solutions. Note that the high-resolution model used to generate datasets is called the *Full Order Model* (FOM) in the ROM framework [31].

The time sequences of multiple computations employing high-fidelity CFD and data-driven ROM are shown in Fig. 1.4. As indicated, the traditional procedure involves a set of high-fidelity calculations for sampled parameters. In comparison, a ROM contains two stages. Firstly, ROM is constructed by the postprocessing of a set of precomputed solutions (e.g., CFD calculations). Then, the reduced system can approximate new data points with a lower computational cost of several orders. The two phases are named *Offline* and *Online*. The unknowns in a ROM are several orders of magnitude fewer than in a FOM. That is indicated in its name of *reduced order*.

High-fidelity ROM approximations within the sampled parametric space are very cheap. Therefore, although the offline step is expensive for a large parameter space, one can still benefit from the cheap online computations and reduce the total time consumed.

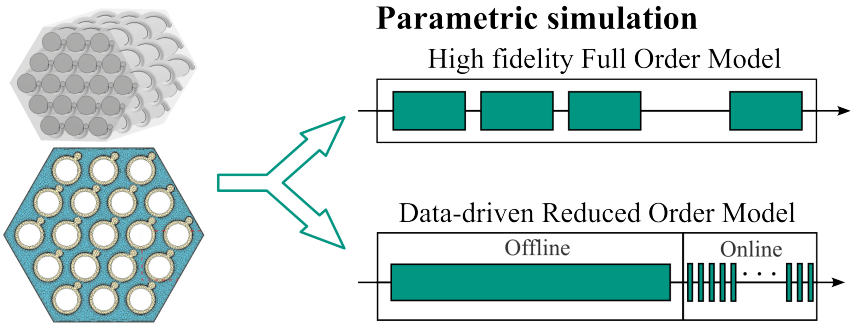


Figure 1.4: Parametric simulation for a typical rod bundle employing high fidelity full order model and/or data-driven reduced order model [32]. Figure (mesh of the bundle) taken with permission from [21], copyright owned by Elsevier.

Similar to the ranking presented in Fig. 1.3, the ROM can also be assessed with respect to cost and accuracy (see Fig. 1.5). As ROM is a data-driven technique built on CFD solutions, its accuracy is slightly reduced (lower than the high-resolution results) [33]. Nevertheless, ROM can reconstruct 3D thermal-hydraulic fields with the same spatial resolution as traditional CFD, which makes it more favorable for reactor analysis compared to lumped parameter approaches.

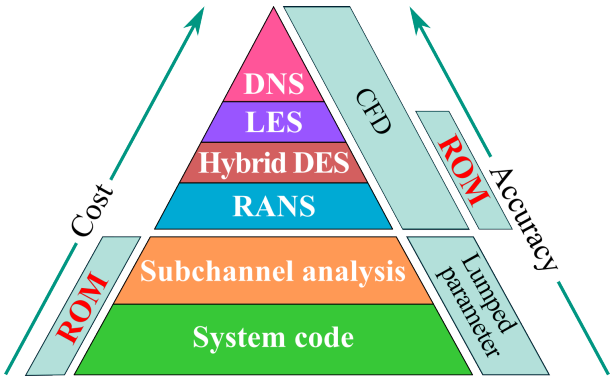


Figure 1.5: ROM in terms of computational cost and accuracy.

Regarding the ranking of the ROM methodology, the next concern is how to quantify its accuracy and reliability. *Verification and validation* (V&V) are regarded as the general means to quantify

these two aspects of simulation techniques [2]. As shown in Fig. 1.6, the CFD and the relevant turbulence model/resolve techniques can be validated through experiments. In addition, the high-accuracy DNS/LES can be adopted to quantify RANS-based models [36; 37].

The V&V strategy also applies to ROMs. Since a ROM is constructed from its corresponding CFD scheme, the standard procedure involves validating the approximations against high-fidelity solutions (see Fig. 1.6). This indicates that CFD solutions are regarded as "true values" for validating ROM approximations.

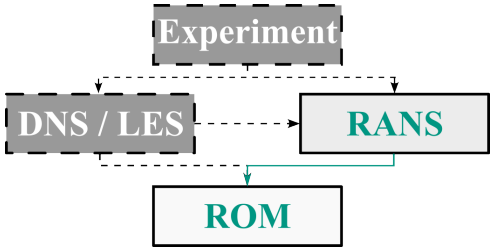


Figure 1.6: Validation procedures of the numerical methods. In this dissertation, ROMs are evaluated by the corresponding RANS solutions [25].

In short, the accuracy of the ROM can be implicitly validated by reliable CFD solutions. However, a comprehensive V&V of the CFD model falls beyond the scope of this dissertation. Thus, the high-fidelity simulations are carefully configured based on recommendations from existing references without specific validation. The configurations include the solver, turbulence model, near-wall treatment, discretization, solution schemes, and mesh. A more thorough discussion about CFD settings is provided in Chapter 3.

1.4 Local ROM: a combination of domain decomposition and Reduced order model

ROMs are efficient techniques for parametric analyses of industrial applications. However, the offline stage tends to be computationally expensive due to the complexity and large scale of nuclear facilities. Consequently, approaches to accelerate the offline phase are important.

To address this challenge, let us recall the philosophy of the traditional *subchannel* approach employed in nuclear engineering, as introduced in Sec. 1.2. Due to the existence of repeating geometrical features in bundles, the entire geometry can be divided in both longitudinal and transverse directions into pieces (i.e., subchannels), as shown in Fig. 1.7. Each part represents a lumped parameter control volume with a single averaged value for each physical quantity.

The same methodology can be applied to construct ROMs for rod bundles. This can be achieved by integrating the *domain decomposition* strategy into the ROM framework. First, the 3D bundle

[2] Readers can turn to the publications [34; 35] for a comprehensive guide to the V&V procedure in CFD.

model is partitioned into multiple subdomains. Then, FOM computations can be performed at the subdomain level, either in several generic partitions or in a small system containing several partitions. ROMs are also built locally. In the online stage, multiple ROMs are assembled to achieve a global approximation.

Note that a ROM based on domain decomposition and the coupling of local predictions is called a *local ROM*. In contrast, the aforementioned strategy for the entire domain is referred to as a *global ROM*. The following paragraphs will illustrate the procedure for decomposing bundles and discuss the challenges and opportunities of using the local ROM approach.

Fig. 1.7 illustrates the procedure of subchannel codes to decompose a seven-pin bare-rod bundle. The domain is divided longitudinally into subdomains of specific length [3]. In the transverse direction, the geometry can be further split into three types of subchannels. The treatment is also suitable for the local ROM construction.

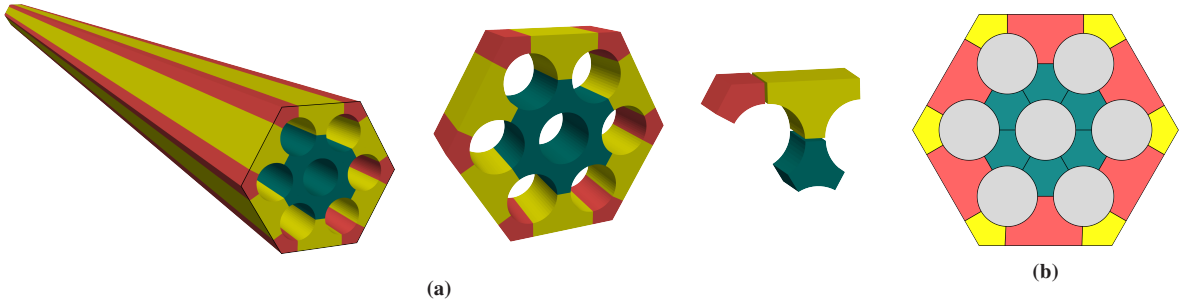


Figure 1.7: Decomposition of a seven-pin bare rod bundle. ((a)) Longitudinal sections and (b) subchannels.

For the wire-wrapped bundle, although the existence of wires increases the geometrical complexity, the model can still be split into repeat partitions. The example displayed in Fig. 1.8 shows two levels of decomposition. Firstly, a long bundle can be divided into subdomains with a length of one wire pitch [4]. Then, each section can be further split longitudinally into six identical parts.

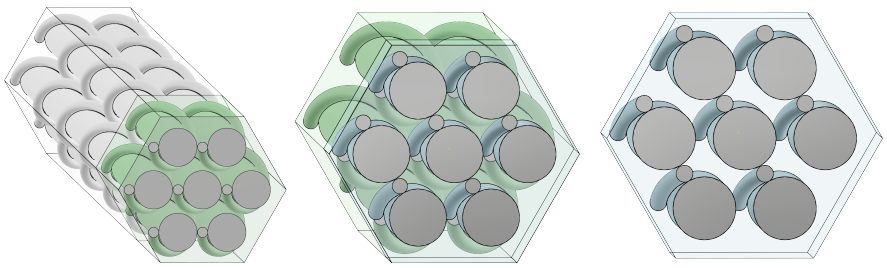


Figure 1.8: Decomposition of a seven-pin wire-wrapped bundle along both the longitudinal direction.

[3] The length of the subdomains should be decided practically, using numerical experiments and evaluated by experimental data.

[4] One wire pitch is the axial distance traveled along the rod after the wire completes a helical turn around it.

Note that subchannel codes can only result in low-resolution approximations due to the coarse spatial discretization. In contrast, the local ROM can reconstruct high-fidelity solutions within each subdomain, providing more informative and accurate results. Thus, the local ROM is a promising technique for modeling large-scale rod bundles, which follows a similar philosophy as the subchannel approach but with a much higher resolution.

However, this coupling introduces challenges. The strategies for domain decomposition and generating representative snapshots must be carefully investigated. More importantly, the assembled problem must satisfy not only the governing equations and constraints of each subdomain but also the interface conditions between adjacent subdomains. Ensuring that local ROMs meet these interface conditions is complex. This topic is currently under active research in the applied mathematics community (see the literature reviews in Sec. 2.4).

1.5 Aims and outline

1.5.1 Motivations

The background and the basics of the techniques adopted in this work are addressed in previous sections. The motivations and objectives of this research are briefly outlined here.

1. Lumpeed parameter methods can be used for modeling of transient scenarios and system-level responses in nuclear reactors. However, their solutions are low-resolution approximations, which are not sufficient for further optimization and analysis.
2. High-fidelity CFD can capture the localized flow and heat transfer phenomena. However, CFD are prohibitively expensive for parameter exploration required for design optimization, uncertainty quantification, and safety analysis.
3. ROMs based on using CFD datasets are efficient for parametric studies. However, most existing work focuses on 2D benchmark cases or small-scale problems. The capability of ROMs for large-scale engineering applications, e.g., rod bundles, is yet to be evaluated.
4. The existence of repeating geometrical features in the rod bundle makes it possible to combine domain decomposition and ROM techniques, indeed, local ROM. However, studies and implementations of local ROMs remain limited for large-scale 3D problems.

1.5.2 Significance and prospects

To clarify the significance of this work, let us revisit the timeline of numerical analysis tools for reactor design, as illustrated in Fig. 1.9. In the early stages with limited computational resources, lumped parameter methods, i.e., system and subchannel codes, have been standard

tools for decades since the 1960s and 1970s [13; 14]. High-fidelity CFD approaches, i.e., finite volume-based RANS models, have gained popularity since the 1990s.

To address the challenges of parametric simulations, reduced order models based on rigorous mathematical theories (e.g., the reduced basis method) have matured since 2000. As repeating geometrical patterns are prevalent in various application problems, the local ROM, become a promising method since the 2010s. Note that, recently, machine learning has been integrated into the ROM framework to enhance its capability and flexibility.

Given existing research and industrial demand, the significance of this work lies in evaluating the capabilities of both global and local ROMs for large-scale rod bundles. The successful implementation of these techniques will manifest their potential for engineering applications.

Moreover, the roadmap of this study shows great potential. Based on the existing investigations and achievements of this study, future directions and prospects can be developed in two ways:

1. The local ROM framework can be regarded as a *high-resolution extension of the subchannel approach*. It can be developed following the same procedure as the subchannel method: (i) constructing local ROMs for bundle sections; (ii) obtaining solutions for an assembly; (iii) assembling fuel assemblies to form an entire core.
2. Continue the development of the global ROM. To achieve simulations for detailed industrial models, technical improvements are required. For instance, code development to handle large datasets, enhancements to ROM accuracy and stability, etc.

Note that the implementation of ROM is in its early stages and has the potential to become an integral part of the design workflow at all levels. Therefore, ROMs have other common applications in nuclear engineering:

1. Geometrical parametrization, e.g., varying the rod spacing, wire pitch, etc.
2. Multi-physics problems, e.g., conjugate heat transfer, fluid-structure interaction, thermal-hydraulics-neutronics coupling, etc.
3. ROMs for other reactor system components, e.g., heat exchangers, steam generators, etc.

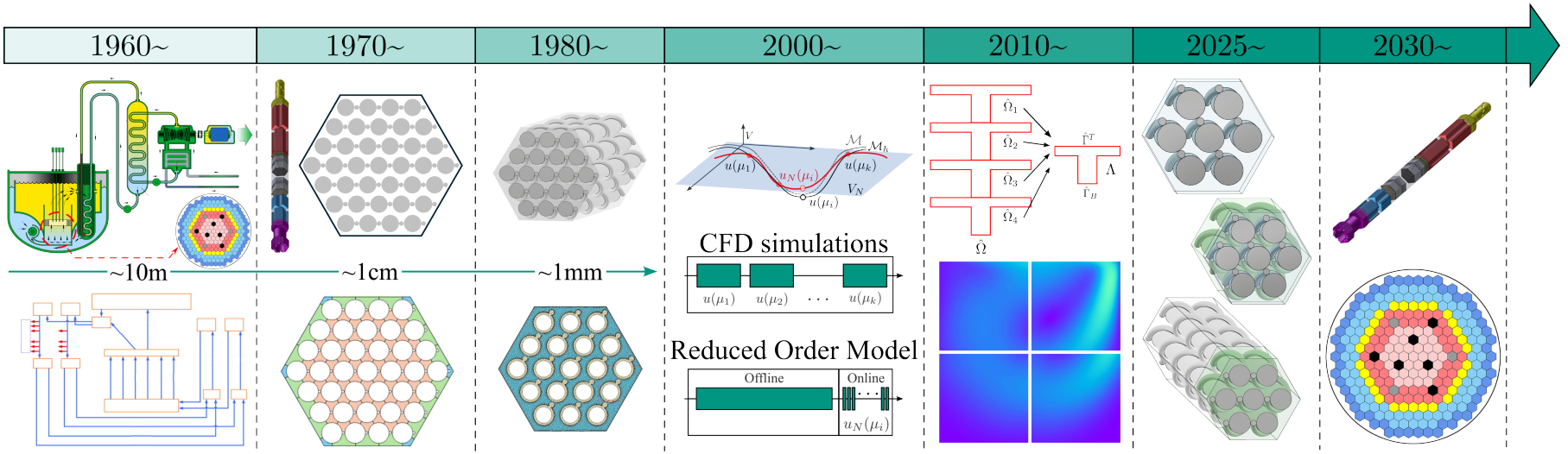


Figure 1.9: A summarized and prospective timeline of the development of numerical tools for the nuclear industry. Conventional methods, i.e., lumped parameter and CFD approaches, have been widely used for several decades. Reduced order models are emerging techniques for parametric simulations. The local ROM is developing rapidly and is expected to be a promising tool. This dissertation focuses on evaluating the capability of both global and local ROMs for typical rod bundles. In the foreseeable future, it will be considered a tool for large-scale reactor modeling. Figures taken with permission from [12; 17; 18; 11; 19; 21; 38; 39; 20], copyright owned by Elsevier.

1.5.3 Tasks and outline

After clarifying the motivations and significance, the next step is to define the goal of this dissertation.

Both intrusive and non-intrusive global ROMs will be tested for the parametric analyses of flow dynamics in rod bundles. The local ROM will also be investigated to assess its potential for practical relevant applications involving repeating geometrical patterns. To achieve the goal, three main tasks are defined as follows.

1. Literature review: The state of the art of CFD investigations for nuclear engineering is reviewed. References are presented and summarized regarding the status of the ROM technique. The characteristics of available methods are discussed, and the selection of approaches for this work is justified.
2. Methodology: The fundamental theories of the approaches applied in this work are introduced in detail, including the high-fidelity CFD and the ROM techniques.
3. Applications: The proposed methods are exploited to analyze flow in typical rod bundles. Both global and local ROM strategies are utilized for physical parametrization (i.e., inflow profiles and viscosity). The global ROM results are shown first, followed by the more challenging local ROM approach.

Be aware that, as ROMs are relatively new to nuclear engineering, this work intends to evaluate their capability for standard physical parametrization, rather than the design scenarios.

The dissertation is structured according to the tasks defined above, as illustrated in Fig. 1.10. The contents of each chapter are briefly described below.

Chapter 2 presents fundamentals and the state of the art in numerical tools, focusing primarily on CFD approaches, ROM methodologies, and the local ROM strategy. The basic principles and mathematical theories are given first. Then, references corresponding to different aspects are reviewed.

In Chapter 3, the methods employed for this dissertation are explained. The RANS formulations and the CFD solver (i.e., software) are described. Both global and local ROM strategies are introduced.

The aforementioned techniques are applied to Global FOM with physical parametrization (see Chapter 4). Three geometries are analyzed: a subchannel, a bare-rod bundle, and a wire-wrapped bundle. The physical parameters include inflow profiles and viscosity.

The local ROM based on the PODI method and the Dirichlet-Neumann iteration is applied to model the wire-wrapped bundle, in Chapter 5. The same set of physical parameters as in the previous chapter are investigated. The performance of the local ROM is systematically assessed.

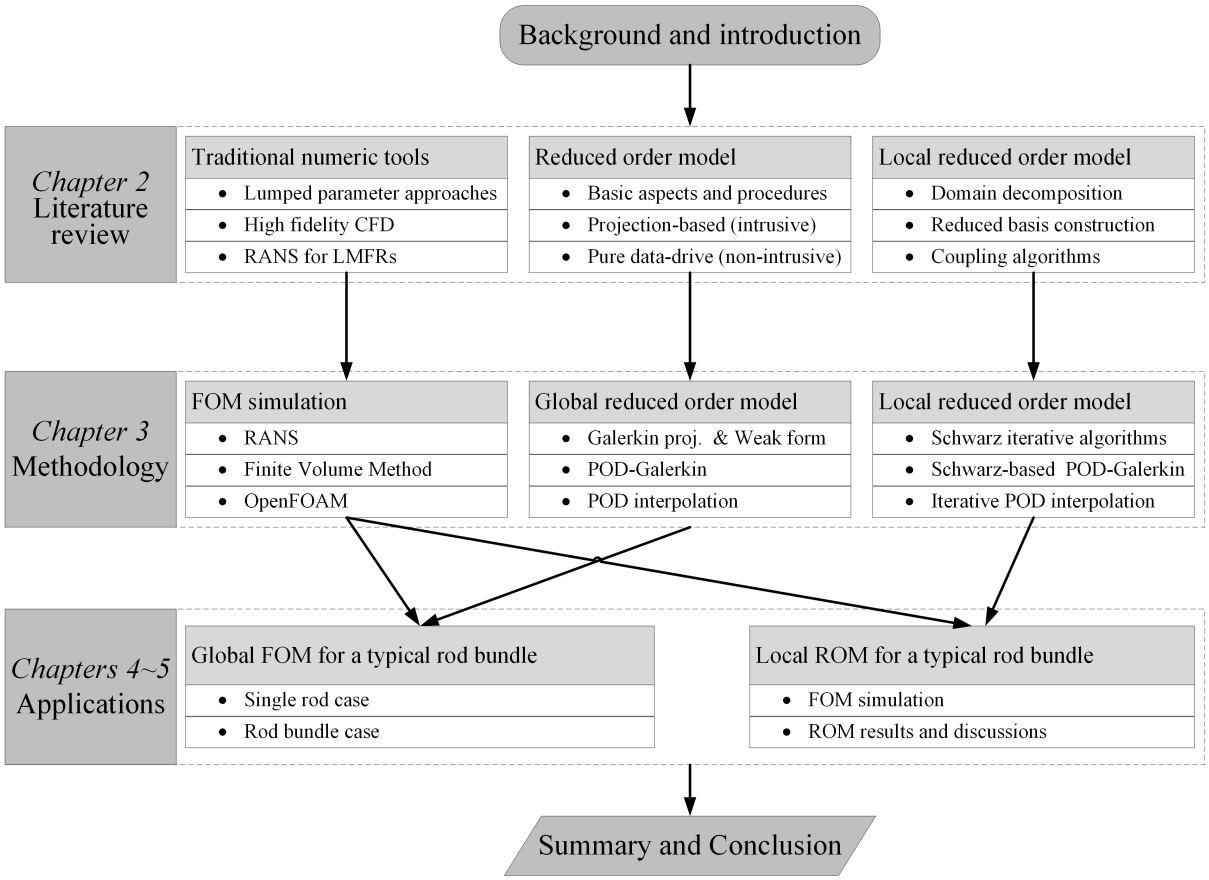


Figure 1.10: Structure of the dissertation. The arrows indicate the connections among sections, e.g., the relations between methods and applications.

The achievements and results are summarized and concluded in the final (Chapter 6), along with perspectives on the application of ROM to these engineering problems.

2 Fundamentals and literature review

Numerical simulation is becoming a standard approach for designing and optimizing various industrial applications. Various tools have been proposed to capture physical phenomena at different length scales in nuclear reactors. The following sections will describe the techniques and outline the state of the art, with a focus on thermal-hydraulic analysis in nuclear engineering.

This chapter is organized as follows. The *lumped parameter* methods are still briefly described. Regarding high-fidelity CFD, a general description and three major approaches for turbulence modeling are discussed. As the number of practices and publications using CFD for numerical analysis is enormous, only CFD modeling for reactors is briefly reviewed.

Recently, ROM has gained popularity due to its advantages for parametric calculations. Two major categories of ROM have been proposed over the past decades: intrusive and non-intrusive. The basic ideology of both approaches is introduced, along with an overview of relevant publications.

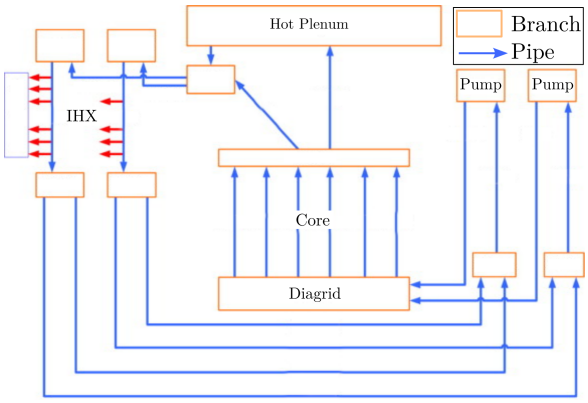
Due to the complexity of industrial problems. The generation of high-fidelity solutions for the entire domain becomes a bottleneck in the implementation of ROM. Fortunately, various engineering facilities are designed to contain repeating geometric patterns. Hence, domain decomposition can be integrated into the framework of ROM, which is known as local ROM. That denotes collecting snapshots only from small-scale subdomains, which is more efficient. Then, the local reduced bases are *glued* to reconstruct global fields. Considering the potential of local ROM for industry related applications, a comprehensive introduction of references is presented. Various methods are classified and summarized. They are mainly evaluated based on an engineering perspective.

2.1 Lumped parameter methods

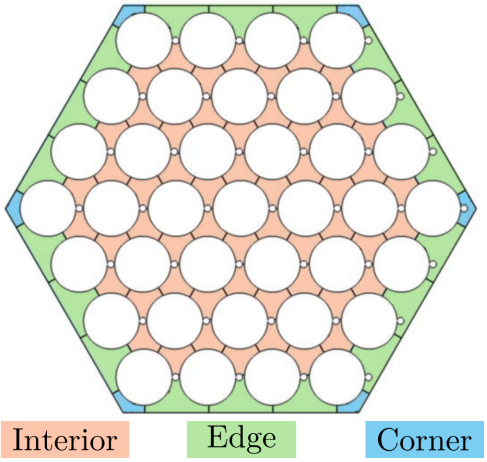
Hardware capabilities have grown dramatically over the past few years. However, due to the large scale of nuclear reactors, *lumped parameter* approaches are still necessary to analyze system-level heat and fluid transport processes. The two types of lumped parameter software (see Fig. 1.2), namely 1D *System* codes and quasi-3D *Subchannel* analyses, will be discussed below.

Lumped parameter models are constructed under the assumption that a system (or component) is divided into homogenized control volumes. In every volume, one averaged value of each physical quantity is defined, and spatial gradients of variables do not exist. Both the one-dimensional

(1D) *System* approach and the quasi-3D *Subchannel* analysis are based on the lumped parameter framework.



(a) System nodalization.



(b) Subchannel discretization.

Figure 2.1: Discretization of a liquid sodium cooled fast reactor by the two lumped parameter tools. (a) The whole system is roughly modeled with a set of control volumes in system codes, taken with permission from [18], copyright owned by Elsevier. (b) A typical 37-pin bundle is discretized into subchannels (interior, edge and corner), taken with permission from [20], copyright owned by AIP Publishing.

The system code adopts control volumes to model the entire reactor system (see Fig. 2.1a). All complex components are simplified and represented as volumes with a single value for each physical quantity (e.g., pressure, temperature, velocity, etc.). The subchannel approach employs the same discretization strategy for rod bundles (see Fig. 2.1b). Nevertheless, those techniques have two major shortcomings. First, the coarse discretization cannot resolve local flow (e.g., in real 3D scenarios). Second, the simplification requires empirical correlations to approximate essential phenomena, such as flow resistance, wall-fluid heat transfer, two-phase flow, etc. Despite the limitations, the system and subchannel codes are considered the workhorse used to assess integral behavior and transient performance of most existing reactors since the 1970s. Recently, their drawbacks have encouraged the research community to adopt contemporary tools, e.g., high-fidelity CFD.

More details regarding them are presented in Appendix A.1.1. The interested reader can find in-depth descriptions from the review articles [40; 41; 42; 16; 43; 25; 44].

2.2 High fidelity computational fluid dynamics

High-fidelity CFD aims to approximate the numerical solutions of *Partial Differential Equations* (PDEs). That denotes field values over the computational domain at finite discrete points with high resolution. Indeed, discretization in both spatial and temporal space should be fine enough to obtain reasonable approximations [45]. Nowadays, CFD is gaining popularity as computational power increases. This section describes several key aspects of implementing CFD, particularly

its applications in nuclear engineering. The interested reader can also refer to [46] for an overview of the topic.

2.2.1 Basics of CFD

In general, a simulation using CFD consists of two major steps [24]:

- Domain discretization. The fluid domain is divided into non-overlapping cells or elements, yielding a grid or mesh system. The subdivision of the computational domain is sufficiently fine. Thus, it can resolve delicate local flow patterns. Moreover, in transient problems, the temporal space is also divided into many time steps.
- Equation discretization. The governing PDEs are discretized into a set of algebraic equations with respect to the mesh and time steps. The algebraic equations in each grid and time point are then assembled into a global matrix. The algebraic system is solved iteratively to achieve the final solutions.

The mesh generation methods and grid shapes are not unique. They need to be adjusted to well represent geometries and satisfy the requirements of discretization schemes, solution algorithms, etc. For more details about *meshing*, the reader may refer to [47].

The algebraic system can be constructed in frames of different approaches, including *Spectral Element* (SEM) [48], *Finite Element* (FEM) [49], *Finite Volume* (FVM) [24], *Discontinuous Galerkin* (DG) [50], etc. These approaches yield matrices with distinct characteristics, and different algorithms are developed to address the problems.

FVM is regarded as the most practical method in modeling engineering fluid flow problems and related transport phenomena. The early work of development was done by Spalding et al. [51]. Also, contributions from Patankar, Gosman, and Runchal [52; 53; 54] are essential. Since then, it has taken a prominent role in CFD due to two significant advantages. Firstly, the dependent variables, i.e., mass, momentum, and energy, are conserved at the local level. Secondly, the integration form of FVM can approximate complex geometries with arbitrary meshes [24].

2.2.2 CFD and turbulence modeling

In engineering applications, the vast majority of flows are turbulent, which is of great practical importance for the transport and mixing of matter, momentum, and heat in fluids. As the turbulent flow fluctuates and consists of multi-scale flow coherent structures, several numerical approaches have been proposed to represent them [22] (see Fig. 1.3).

The main advantages of CFD over lumped parameter methods are its higher resolution and ability to model (or resolve) turbulence. The four most widely used approaches, i.e., DNS, LES, DES, and RANS, are illustrated in Fig. 1.3, which vary in computational cost and complexity. All

of them can be incorporated with FVM. Their theoretical foundations, governing equations, and applications are detailed in [22; 23; 24]. Below, we briefly summarize their general assumptions and key characteristics, excluding DES, which is a hybrid of RANS and LES. Note that the application of interest is the LMFRs, where the fluid is in liquid state, so the studies presented below consider only the single-phase flow.

RANS, LES, and DNS can model the turbulent eddies on different scales. RANS uses time-averaged equations to represent the mean flow field. The turbulent features are approximated with semi-empirical models. LES can resolve large-scale eddies, while the rest, commonly termed *Sub-grid-scale* (SGS) stresses, are also represented with empirical turbulence models. DNS is a fully resolved method. It utilizes a sufficiently fine spatial mesh and small time steps to capture the smallest eddies and the fastest fluctuations in a transient solution. The accuracy of the three methods increases in order, as does the computational cost, as displayed previously in Fig. 1.3.

All the three aforementioned techniques have been widely applied to simulate the fluid dynamics in nuclear engineering [55]. Interested readers can refer to the overview documents [26; 55; 36] for more details. Also, a brief literature review is addressed in Appendix A.1.2. The applications of the three methods for LMFRs are summarized as follows:

- RANS can simulate various problems for large bundles, including thermal-hydraulic assessments [56; 57; 58], flow blockages [59; 60; 61], rod deformation [62; 63; 64], etc.
- LES is employed for thermal-hydraulic modeling at the subchannel level or for typical bundles. See more in the references [65; 66; 67; 68; 69; 70].
- DNS is used for capturing small-scale flow variances in subchannels or even smaller geometries [71; 72; 73; 74; 37; 75].

Despite improvements in hardware over the last few decades, DNS and LES are still prohibitively expensive for most research groups. RANS-based CFD remains the most practical tool for thermal-hydraulic analysis of LMFRs. Nevertheless, even RANS-based approaches lack the efficiency for comprehensive parametrization and real-time calculation/control applications. This situation is aggravated in simulations of reactor fuel assemblies, which contain micro- and macro-structures in a wide range from meters to millimeters. Consequently, the number of grids for a high-quality mesh of the fuel bundle is enormous, which requires huge computational resources for simulations.

To address the limitations of CFD, the research community has invested heavily in developing highly efficient, accuracy-preserving parametrization techniques. Among the alternative methodologies, for its relevance in our study, we highlight *Reduced Order Models* (ROMs).

2.3 Reduced order models

ROM can be regarded as a data-driven method. It aims to obtain high-fidelity solutions with much lower numerical complexity. To be more precise, the unknowns (also known as *Degrees of Freedom* (DoFs)) of a reduced system are much smaller than the original discretized PDEs. Note that, in this context, the original problem is named *Full Order Model* (FOM) in the frame of ROM. In this chapter, the procedures and several essential aspects for ROM are discussed.

2.3.1 Basic aspects

The fundamental goal of ROM is to reduce the computational cost of parametric analysis using existing datasets. In Fig. 2.2, the runtime costs of multiple FOM and ROM calculations are sketched. Suppose one needs to carry out computations corresponding to k parameters, i.e., $\mu_1, \cdots, \mu_k$. It requires running the FOM for k times. The time consumed for every calculation is almost the same. Thus, it is expensive for a large parameter space.

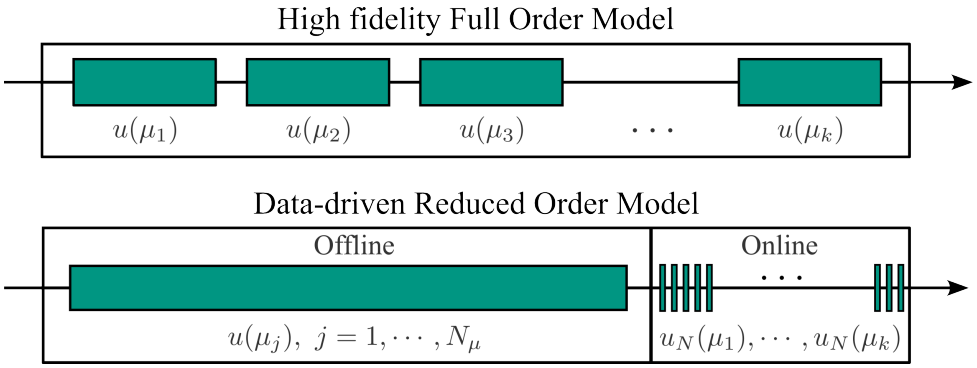


Figure 2.2: Computational consumption of the FOM and ROM for multiple parametric simulations. Figure redraw refers to [32].

The procedure of a ROM for the same parametric study is different. It typically consists of two stages, i.e., offline and online [76].

- For the offline phase, a few parameters $\mu_1, \ldots, \mu_{N_\mu}$ are sampled from the whole parameter space. Note that N_μ is generally much smaller than k. The parameterized solutions u are computed and collected as a dataset called *snapshots*, $\mathbf{S} = \left[u_1, \ldots, u_{N_\mu}\right]$. The matrix $\mathbf{S}$ has a shape of $N_h \times N_\mu$, where N_h denotes the dimension of high fidelity solutions. Then, the data and governing equations are post-processed to build a ROM. Although the offline stage requires a huge computational expense, it only needs to be run once.
- In the online stage, in which the ROM is available, calculations can be carried out cheaply with high accuracy. One can now obtain $u_N(\mu_1), \ldots, u_N(\mu_k)$. Moreover, the ROM can be applied to efficiently obtain solutions within the parameter space, which results in a speedup of several orders of magnitude [76]. The ROM fields have the same dimensions as

the original FOM. These make the approaches prone to carrying out extensive parametric studies and uncertainty quantification [77; 78; 79; 80].

"Note that the strategy to create ROM is not unique. Over the past two decades, various ROM techniques have been proposed, which can be divided into two major groups: *intrusive* and *non-intrusive* approaches. The former requires snapshots and the manipulation of the governing PDEs, whereas the second is purely data-driven, adopting only high-fidelity solutions [81; 82]."[1] The two frameworks and references are overviewed in Sections 2.3.3 and 2.3.4, respectively.

As ROM is not commonly adopted for thermal-hydraulic analysis of nuclear reactors, references from other fields are cited for clarification. An overview of ROM in nuclear engineering is described separately in Section 2.3.5.

2.3.2 Parametrization techniques

Note that before constructing a ROM, the parameters for generating snapshots should be determined. The selection of parameters is crucial to the performance of ROM. The parameterization technique aims to optimally sample values to balance the cost and accuracy of ROMs. Three relevant aspects are included: sampling strategies, geometric parameterization, and parameter space reduction. Although a comprehensive parametrization study is not adopted in this investigation, as an essential part of ROM, it is discussed for completeness.

2.3.2.1 Sampling strategies

Sampling denotes selecting a subset that can well approximate the features of the whole dataset [83]. Sampling strategies vary in different applications, which depends mostly on the dimension, $N_{\mathcal{P}}$, of the parameter space $\mathcal{P}$ [84; 30].

- "For low dimensional (typically $N_{\mathcal{P}} \leq 3$) situations, *grid-based* approaches are the standard option [85; 30].
- When the dimension $N_{\mathcal{P}}$ is between 4 and 10, the number of grid points grows exponentially. Statistical algorithms like random (e.g., *Monte Carlo*) and *Latin hypercube* sampling, as well as Smolyak sparse grid, are well-suited for these conditions [86; 85; 30].
- When the dimension $N_{\mathcal{P}} > 10$, the common sampling techniques mentioned above are not feasible. Therefore, more sophisticated problem-specific *adaptive algorithms* should be utilized, see [85; 84; 87]. The *greedy method* is a widely used adaptive approach for this task [88; 33; 89]."[2]

[1] rephrased from Ruan et al. [2], page 3, 5th paragraph.
[2] rephrased from Ruan et al. [2], page 10, the last paragraph.

2.3.2.2 Geometrical parametrization

It is straightforward that physical quantities, e.g., viscosity, boundary conditions, source terms, etc., can be regarded as parameters. Actually, in many engineering applications, the goal is to design and optimize the shape of a component or analyze the dynamics in different geometries [90; 91; 92]. In these cases, geometrical parametrization is necessary. See more applications in [88; 30; 89; 93; 94; 31].

Through a literature survey, we identified two main categories of geometrical parametrization methods: *interpolation technique* and *Free Form Deformation* (FFD). The implementation of both approaches requires a reference geometry and mesh. Their procedures and observed references are listed below.

The interpolation technique is a straightforward method to generate new geometries. It is based on the interpolation of the reference geometry and mesh. The parameters are defined as the coefficients of interpolation. For example, if we want to generate a new geometry with a length of L, we can define a parameter μ as the ratio of L to the length of the reference geometry, i.e., $\mu = L/L_{\text{ref}}$. Then, the new geometry can be obtained by interpolating the reference one with μ. The method is simple and efficient, but it is limited to specific types of geometrical deformation.

FFD is a general deformation method for 2D and 3D objects, which is independent of the geometry and mesh [95; 96]. It can also be used for the geometrical parametrization of a part of a domain [91; 97]. It consists of four steps: (i) the reference domain is embedded in a lattice of control points; (ii) the control points are moved; (iii) the new geometry is obtained through interpolation of the deformed lattice; (iv) the mesh is generated through interpolation of the new domain. Note that the mesh can also be deformed directly according to the movements of control points, and the procedure is simplified to three steps. By employing FFD, displacements of the control points are parameterized.

2.3.2.3 Parameter space reduction

The last aspect of parametrization is the reduction of the parameter space. It is proposed to deal with problems with a huge number of parameters (i.e., high-dimensional parameter space). For industrial scenarios, the design and optimization of a device might involve numerous factors. That results in difficulties in both FOM and ROM simulations. The aim is to reduce the dimension of the parameter space while preserving the main features of the original one [84]. Additionally, low-dimensional inputs also improve the accuracy of ROMs, as demonstrated in [98; 99; 100; 101; 102].

Now, two major approaches will be briefly introduced, i.e., *Active subspaces* and the *AI-based generative model*.

Active subspace The *active subspace* (AS) method can generate several *active* values that represent linear combinations of the original parameters. Then, the low-dimensional active values are adopted as new parameters [103].

For example, assume a parameter space $\mathcal{P}$ has $N_{\mathcal{P}}$ dimensions, a point in $\mathcal{P}$ can be defined as $\boldsymbol{\mu} = [\mu_1, \ldots, \mu_{N_{\mathcal{P}}}]^{\mathrm{T}} \in \mathbb{R}^{N_{\mathcal{P}}}$. The AS method can compute a linear transformation matrix $\mathbf{W}_1 \in \mathbb{R}^{N_{\mathcal{P}} \times M}$, where $M < N_{\mathcal{P}}$. The original parameters are projected to a new space with a reduced dimension, i.e., $\boldsymbol{\mu}_M = \mathbf{W}_1^{\mathrm{T}} \boldsymbol{\mu} \in \mathbb{R}^M$, and $\boldsymbol{\mu}_M$ are called active variables.

The method was initially proposed by Paul G. Constantine et al. [104] and has been applied in various applications, including airfoil [99], ship hull [100; 105; 101], and ship propeller blade optimizations [106].

Generative model The *generative model* for parameter space reduction is a state-of-the-art approach. It is a neural network trained to learn the distribution of the original parameters. Then, the network can generate new samples that follow the same distribution, but with lower dimensionality [101].

The basic ideology of the generative model is similar to the active subspace method. The key difference is that the transformation matrix $\mathbf{W}_1$ is now replaced by a neural network. For example, let us assume a network as $\mathcal{Z}$ and a point in the original parameter space as $\boldsymbol{\mu}$, the corresponding low-dimensional value can be represented as $\boldsymbol{\mu}_M = \mathcal{Z}(\boldsymbol{\mu})$.

The incorporation of the generative model and ROM can be seen in [101; 107; 108]. It has been applied to several scenarios, including the Poisson problem and Reynolds-Averaged Navier-Stokes equations. The authors have show that the framework is more efficient and accurate than one without parameter space reduction.

2.3.3 Intrusive Reduced Basis method

Now, let us begin the discussion of ROM construction using an intrusive framework. Fundamental theories and applications of intrusive techniques have been summarized in the books [88; 33; 89; 93; 76]. The following sections introduce a relavant reduced-order modeling technique: the *Reduced Basis* (RB) method.

Its construction contains two parts: (i) extracting a reduced basis from high-fidelity solutions; (ii) employing the *Galerkin* method to project the original governing equations onto a subspace spanned by the reduced basis [30]. Note that, as the formulations are derived using Galerkin projection, it is also called a projection-based method.

Fig. 2.3 illustrates the basic principle of the reduced basis method. Assume the solution of a PDE exists within a specific function space denoted $\mathcal{V}$. The true solution set $\mathcal{M}$ is represented by a high-fidelity database $\mathcal{M}_h$. $u_h(\mu_1), \ldots, u_h(\mu_k) \in \mathcal{M}_h \in \mathbb{R}^{N_h}$ are a set of FOM snapshots.

Then, a reduced subspace $\mathcal{V}_N$ is computed from $\mathcal{M}_h$ via a dimensionality reduction technique. A ROM is built on $\mathcal{V}_N$ to approximate $\mathcal{M}_h$. The dimension of $\mathcal{V}_N$ is N^{RB}, which is much smaller than N_h. The ROM approximation $u_N(\mu_j) \in \mathcal{V}_N$ is expected to be close to $u_h(\mu_j)$. The ROM error denotes $\varepsilon_j = \|u_h(\mu_j) - u_N(\mu_j)\|$.

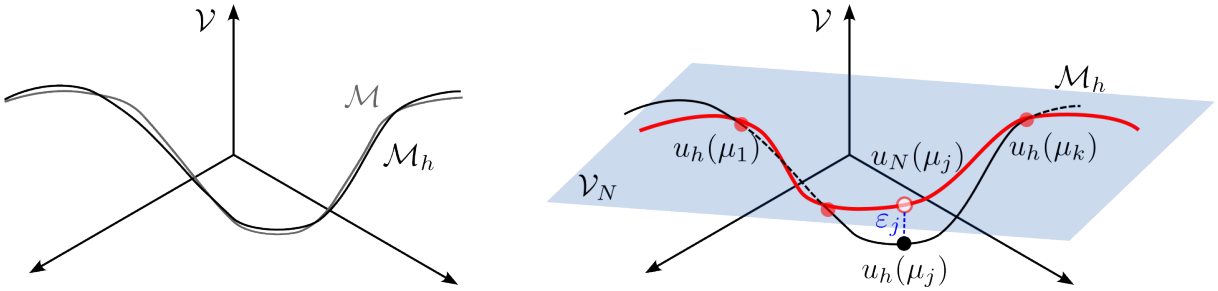


Figure 2.3: Sketch of the numerical approximation. Left: the true solution set $\mathcal{M}$ is approximated by a high-fidelity set $\mathcal{M}_h$. Right: N^{RB} parametric snapshots are collected to construct a RB $\mathcal{V}_N$, shown as a hyperplane. A ROM result $u_N(\mu_j)$ exists within $\mathcal{V}_N$ and its error to $u_h(\mu_j)$ denotes ε_j.

The following sections describe the two key steps of the RB method, i.e., RB generation and Galerkin projection.

2.3.3.1 RB generation

Suppose a snapshot set $\mathbf{S}$, dimensionality reduction techniques are applied to extract several dominant features (i.e., RB vectors) from $\mathbf{S}$. The RB is spanned by N^{RB} basis vectors, i.e., $\mathcal{V}_N = \text{span}\left\{\mathbf{v}_i, i = 1, \ldots, N^{\text{RB}}\right\}$. Then, $\mathbf{S}$ can be reconstructed by a linear combination of $\mathbf{v}_i$. Indeed, each FOM field can be expressed as $\mathbf{u}(x;\mu_j) \approx \sum_{i=1}^{N^{\text{RB}}} a_i(\mu_j)\mathbf{v}_i(x), \quad \mathbf{v}_i \in \mathcal{V}_N$. The methodology is illustrated by an example displayed in Fig. 2.4.

The algorithm to construct a RB is not unique. Two popular methods are discussed and reviewed in the section, i.e., *Proper Orthogonal Decomposition* (POD) and the *Greedy Basis Generation* algorithm [30; 33; 76]. There are many other dimensionality reduction techniques available, such as *Dynamic Mode Decomposition*, *Linear Discriminant Analysis*, *Proper Generalized Decomposition*, and *Active Subspaces*, etc. But they are not covered in the current discussion.

Proper orthogonal decomposition POD is a widely used dimensionality reduction technique for analyzing complex spatial-temporal systems [109]. The method captures the most essential features of snapshots with a reduced representation. POD modes can be computed using *Singular Value Decomposition* (SVD) or *Eigendecomposition* of the snapshot correlation matrix [30].

POD was initially used to analyze turbulence and coherent structures in fluid dynamics [110; 111]. The applications suggested that POD is an effective method to interpret complex data. Then, it was extended to extract dominant features of various physical fields, including pressure, velocity,

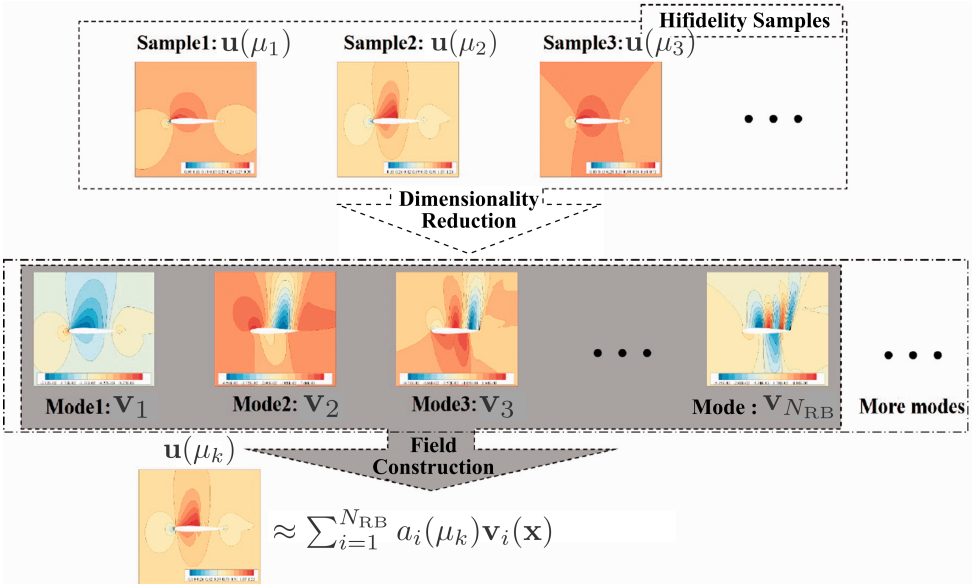


Figure 2.4: Illustration of the reduced basis approximation. Dominant features $\mathbf{v}_i$ are extracted from snapshots $S = \left[\mathbf{u}(\mu_1), \ldots, \mathbf{u}(\mu_{N_\mu})\right]$. High resolution fields can be reconstructed considering a linear combination of basis functions. Figure taken with permission from [82], copywriter owned by Institution of Mechanical Engineers.

and temperature in fluid dynamics [112; 113], displacement and stress in structural analysis [114], etc.

In the frame of ROM, POD plays a leading role in the post-processing of data sets, including transient and other parametric simulations [30].

For time-dependent applications, snapshots are solutions taken at a series of times. POD is then used to extract the essential dynamics during the variations [112; 115; 77; 116; 78].

Recently, POD has been applied to more general conditions, indeed, the parametrization of physical and geometrical variables. Various physical values can be considered, such as fluid viscosity, boundary velocity, etc [115; 117; 118; 113; 119]. Readers can also refer to the studies [91; 94; 120; 107] that regard deformations of the computational domain. Additionally, POD has shown its capability in dimensionality reduction for a wide variety of PDEs, such as elliptic [38], parabolic [121], hyperbolic [122], Stokes [123], and Navier-Stokes [113], etc.

In this research, POD is utilized for reduced basis construction, and its formulations are presented in Sec. 3.2.1.2.

Greedy algorithm The greedy algorithm is another widely used method to generate a RB [30; 33]. It is an iterative procedure that selects the most relevant snapshots (or parameters) by optimizing a specific criterion. In each step, a new parameter is identified, and the corresponding high-fidelity solution is obtained. A new basis function is then computed by employing orthogonalization (e.g., *Gram-Schmidt orthogonalization*) [124]. The newly expanded basis functions can optimize the overall representation of the RB to the problem [30]. The procedure continues

until reaching a predefined tolerance. Hence, the dimension of the reduced subspace equals the number of solutions.

The basis enrichment can be achieved through two distinct procedures, depending on whether FOM solutions are available [30].

The first is called the *strong* greedy algorithm [30]. It is employed when not all FOM simulations have already been performed. It is an iterative procedure for selecting parameters and performing computations.

Before the procedure, a set of candidate parameters, $\mathcal{P} = [\mu_1, \ldots, \mu_N]$, is defined. Then, the iterations proceed as follows.

1. An initial parameter μ^0 is selected, and the corresponding high-fidelity solution $\mathbf{u}(\mu^0)$ is computed. The first basis $\mathbf{V}_0 = \text{span}\,\{\mathbf{u}(\mu^0)\}$.
2. A ROM is built from $\mathbf{V}_0$ and applied to approximate solutions for all parameters in $\mathcal{P}$. The next parameter μ^1 is selected by maximizing the error between the ROM and FOM, i.e., $\mu^1 = \arg\max_{\mu_j \in \mathcal{P}} \|\mathbf{u}(\mu_j) - \mathbf{u}_N(\mu_j)\|$. The new snapshot $\mathbf{u}(\mu^1)$ is computed. $\mathbf{u}(\mu^0)$ and $\mathbf{u}(\mu^1)$ are orthogonalized to obtain a new basis vector $\mathbf{V}_1$.
3. The iteration continues and selects $\mu^2, \ldots, \mu^N$. It ends until a predefined convergence tolerance ε is satisfied, i.e., $\max_{\mu_j \in \mathcal{P}} \|\mathbf{u}(\mu_j) - \mathbf{u}_N(\mu_j)\| < \varepsilon$.

Note that the criterion $\|\mathbf{u}(\mu_j) - \mathbf{u}_N(\mu_j)\|$ is not computable, as the FOM solution $\mathbf{u}(\mu_j)$ is unknown. Hence, the value is estimated by a priori *error estimator*, which is a analytical quantity that can bound the true error [30].

The key advantage of this algorithm over POD is that the size of FOM simulations is increased adaptively, if necessary, to enhance accuracy. However, finding an error estimator is an essential question for its implementation (see [125; 124; 126; 127]).

The second procedure is called the *weak* greedy algorithm. It is applied to the case where snapshots have already been generated. The aim is to only adopt a subset of the data for dimensionality reduction [30]. Thus, instead of the error estimator, the selection can be determined by the relative projection error,

$$\mathcal{E}_j^k = \frac{\left\| \mathbf{u}(\mu_j) - \Phi^k {\Phi^k}^{\mathrm{T}} \mathbf{u}(\mu_j) \right\|}{\|\mathbf{u}(\mu_j)\|},$$

where k is the current iteration number, $\Phi = [\mathbf{v}_1, \ldots, \mathbf{v}_{k-1}]$ is the RB matrix. $\mathcal{E}_j$ denotes the relative projection error of the j-th snapshot.

In each iteration, $\mathcal{E}_j^k$ is computed for all snapshots. The $\mu^{n+1} = \arg\max_{\mu_j} \mathcal{E}_j^k$ is selected to enrich the reduced subspace. Similarly, the iteration ends when the maximum $\mathcal{E}_j^k$ is acceptable. It is worth noting that the weak greedy algorithm is not adaptive, as all snapshots are already

available. The weak greedy strategy does not require an error estimator that is difficult to find. However, it is less efficient than the strong one, as all FOM simulations are performed before the selection.

The work on the greedy approach for ROM can be found in the 2000s [128]. A priori convergence and efficiency analysis were provided in [124]. Bernard Haasdonk [126] adopted POD in greedy iterations to compute the reduced basis vectors. The approach has recently been used to construct the RB for various parameter-dependent scenarios, such as the Poisson equation [38], the Burger equation [128], structural dynamics models [129], the Helmholtz problem [130], and Navier Stokes equations [131].

2.3.3.2 Galerkin projection and ROM

The ROM is built upon the RB, and the reduced system is achieved via a well-known technique, *Galerkin projection*. It is a mathematically rigorous method used to reformulate PDEs before constructing algebraic systems. The original PDEs (*strong form*) are transformed into the so-called *weak form* by projection. This is the fundamental procedure for many discretization algorithms, such as the Spectral Method (SM) [48], the Finite Element Method (FEM) [49], the Discontinuous Galerkin (DG) [50], etc.

Strong and weak form The derivation of the weak form of a PDE has been extensively documented in the literature [49; 48; 45]. The formulations are briefly summarized for completeness.

The strong form of a general parametric PDE is expressed as

$$\dot{u}\left(x,t\right)+\mathcal{Q}\left(u\left(x,t\right);\mu\right)=f\left(x,t;\mu\right), \tag{2.1}$$

where $\dot{u}$ is the time derivative of u. $\mathcal{Q}$ is a differential operator, not necessarily linear. f is a given source term. μ denotes parameters. The boundary and initial conditions are not presented for brevity.

By applying the inner product with the so-called test function w, equation (2.1) is reformulated as the *weak form*,

$$\left(\dot{u},w\right)+\mathcal{O}\left(u,w\right)=\mathcal{F}(w), \tag{2.2}$$

where $(\cdot,\cdot)$ denotes the inner product, and w are selected from a test functional space $\mathcal{W}$. For brevity, μ is omitted. The definition of the test space depends on the problem.

Let us expand u with basis functions $v_j \in \mathcal{V}$ (also known as *trial functions*), in which $\mathcal{V}$ is a functional space. Suppose $u \approx u_N = \sum_{j=1}^{N_{\text{RB}}} a_j v_j$, the weak form (2.2) is expressed as finding the approximation $u_N = \sum_{j=1}^{n} a_j v_j$ with $v_j \in \mathcal{V}$ that satisfies

$$\left(\sum_{j=1}^{n} \dot{a}_j v_j, w_i \right) + \mathcal{O} \left(\sum_{j=1}^{n} a_j v_j, w_i \right) = \mathcal{F}(w_i). \tag{2.3}$$

Note that the unknowns of equation (2.3) are the coefficients a_j. Thus, N_{RB} test functions w_i are required to close the system.

"The choice of test functions is crucial for the well-posedness of the reduced system. The test functions are selected from a test space $\mathcal{W}$. The selection can be based on the trial space $\mathcal{V}$, which is spanned by the basis functions v_j. For example, if $\mathcal{W} = \mathcal{V}$, the projection is called the Bubnov-Galerkin projection. If $\mathcal{W} \neq \mathcal{V}$, it is called the Petrov-Galerkin projection. The choice of test functions can affect the stability and convergence of the ROM. Hence, it is an important aspect to consider when constructing a ROM."[3]

A special case of Galerkin ROM is the *Least-Squares Reduced Basis* (LS-RB) method [30]. The approach aims to obtain solutions $u_N \in \mathcal{V}$ satisfying the optimal system (2.1)

$$u_N = \arg\min_{v \in \mathcal{V}} \left\| (\dot{u}, w) + \mathcal{O}\left(u, w\right) - \mathcal{F}(w) \right\|^2, w \in \mathcal{W}. \tag{2.4}$$

Application of projection-based ROM The intrusive ROM has been widely used to address various problems over the last two decades. We will categorize the existing studies considering several aspects, i.e., FOM discretization schemes, model problems, and parameters, and review them separately.

As high-fidelity solutions are obtained utilizing different CFD discretization techniques, the existing studies show that the projection-based ROM works well with the datasets collected from FEM [132; 119], FVM [113; 118], DG [133; 134], etc. A collection of those studies is presented in [76].

They are also capable of solving various PDEs, like the Poisson equation, Burgers' equation, Euler equation, and Stokes equation [123; 115; 32; 77; 80; 107].

Remarkably, the incompressible Navier-Stokes for laminar flow can also be analyzed in the frame of ROMs [118; 116; 78]. However, the basis functions of velocity are divergence-free due to the incompressible assumption, which already satisfies the continuity equation. Hence, the well-posedness (i.e., existence, stability, and convergence of solutions) of the reduced systems is not

[3] rephrased from Ruan et al. [2], page 21, 2nd paragraph.

preserved. Thus, the Poisson equation for pressure [118; 135; 136] or a supremizer enrichment technique [123; 117; 33] is proposed to resolve the issue [4].

Model order reduction methods have already been extensively applied to many scenarios, and various quantities can be regarded as parameters for ROMs, like physical variables (e.g., time, velocity, viscosity, temperature, etc [93; 76; 137]) and geometric deformations [138; 117; 91; 76]

In addition, the presence of nonlinear terms leads to high-dimensional reassembly when solving a reduced system, which weakens the reduction capabilities of the ROM. Consequently, some hyper-reduction techniques have been developed to efficiently deal with the obstacle, such as the empirical interpolation method (EIM) [139], its discrete variant DEIM [140], and its matrix version (MDEIM) [141]. Those approaches are not covered in this document because they are beyond its scope. Interested readers can refer to the cited references.

The intrusive approaches for Navier-Stokes flow have recently been extended to include turbulence models. The approach uses high-fidelity turbulent solutions obtained from CFD simulations using RANS [142; 113] or LES [143; 144]. The continuity and momentum equations are then reduced by projection. The additional terms associated with turbulence models are not fully included in the reduced system. Instead, only the eddy viscosity is considered and interpolated employing surrogate models, e.g., the Radial Basis Function. The turbulent contribution is added as an additional term to the momentum formulation.

This research considers the standard Galerkin ROM for RANS, see Sec. 3.2.2 for details.

2.3.4 Non-intrusive ROM: machine learning

Non-intrusive methods are purely data-driven and typically do not manipulate PDEs. It trains an efficient surrogate model using high-dimensional datasets and can approximate FOM results at low computational cost.

In general, the approaches work in a black-box manner without requiring access to the CFD source code, offering greater flexibility for practical problems than Galerkin-based procedures. They are becoming popular, accompanied by the rapid development of *Machine learning* (ML). Various non-intrusive ROMs have been proposed during the past decade. A recent overview of ML for fluid dynamics can be seen from [145; 76; 146].

The following sections will introduce several widely used techniques for building surrogates for specific inputs and outputs. In addition, how to incorporate the approaches to build ROMs are discussed. Given the relevance to this thesis, two ROM construction methods will be discussed: POD with interpolation and Data-Driven Neural Networks (DDNN).

[4] Supremizer enrichment is a well-known technique, which ensures the reduced system satisfies the inf-sup condition. It is necessary for the existence of a unique solution and the stability of numerical approximations.

2.3.4.1 Proper orthogonal decomposition with interpolation

The first to be mentioned is called *Proper orthogonal decomposition with interpolation* (PODI). As the name implies, POD is applied in the offline phase to reduce the dimensionality of the collected data, extracting the most dominant modes. An interpolation model is subsequently adopted to correlate the input parameters to the corresponding POD coefficients [94; 145]. After training, it can predict coefficients with new inputs and reconstruct the high-dimensional fields.

Assume the parameter space $\mathcal{P}$ and the POD approximation

$$u(\mu_k) \approx \sum_{i=1}^{N^{\mathrm{RB}}} \alpha_i(\mu_k) v_i, \qquad \forall \mu_k \in \mathcal{P}.$$

A surrogate $\mathcal{Z}$ is trained to achieve

$$\alpha_i(\mu_k) \approx \mathcal{Z}(\mu_k),\ \forall \mu_k \in \mathcal{P},\ i = 1, \ldots, N^{\mathrm{RB}}. \tag{2.5}$$

Thus for each new value μ_{new}, one can obtain $\alpha_i(\mu_{\text{new}}) \approx \mathcal{Z}(\mu_{\text{new}})$. The resulting field is reconstructed directly via $u(\mu_{\text{new}}) \approx \sum_{i=1}^{N^{\mathrm{RB}}} \alpha_i(\mu_{\text{new}}) v_i$.

A very early application of PODI (with cubic spline interpolation) is proposed by Bui-Thanh for steady aerodynamics analyses [147]. Various techniques have been adopted to obtain the interpolant $\mathcal{Z}$, including regression algorithms such as linear models, Radial Basis Function (RBF) Interpolation, and Gaussian Process Regression (GPR). Artificial Neural Networks (ANN) are also well-suited for the task.

Due to its simplicity and generality in constructing ROMs, the PODI has been widely adopted for analyzing computational physics applications. A Python package, *EZyRB* [148], is recommended for easy implementation of PODI. Note that it beyond the scope to review numerous existing articles. The following paragraphs highlight only a few typical studies on interpolation algorithms and application scenarios.

Considering the techniques to build $\mathcal{Z}$, there are several possibilities, such as linear algorithm (linear interpolation [149], and Delaunay triangulation and its dual Voronoi tessellation [150]), RBF [151; 152; 153], probabilistic based-approches (Bayesian Kriging [154], GPR [155; 98; 156]), and ANNs [157; 120; 97].

It is also suitable for various application fields, including unsteady Navier-Stokes flow [158; 159], shape optimization with respect to the heat conduction [149], aerodynamics [160], geometrical parameterized [150; 107], Fluid-Structure Interaction (FSI) problems [161; 92], neutron transport [162; 163; 164], and quasi-1D combustion behavior [165].

2.3.4.2 Neural Networks for ROM

The basic ideology of building ROM utilizing neural networks also falls within the framework of offline and online stages. The FOM solutions corresponding to selected parameters are generated. Various frameworks have been proposed to map inputs and outputs. Considering the popularity of neural networks for scientific computation, many books are available [109; 166; 167; 168] for detailed instructions.

As the dimensionality of physical fields is generally huge, it is not feasible to directly surrogate the input parameters and output fields. Therefore, dimensionality reduction techniques are still incorporated to extract the most relavant features. Then, neural networks can be trained to involve the effect of parameters on the desired outputs.

Autoencoder (see Fig. 2.5) is a broadly used network architecture for dimensionality reduction [169]. It contains two parts, *Encoder* and *Decoder*. The former can generate a low-dimensional *latent space* (i.e., $\boldsymbol{\alpha}(\mu)$) from the original snapshots. The latter can reconstruct the high-dimensional fields from the low-dimensional representation.

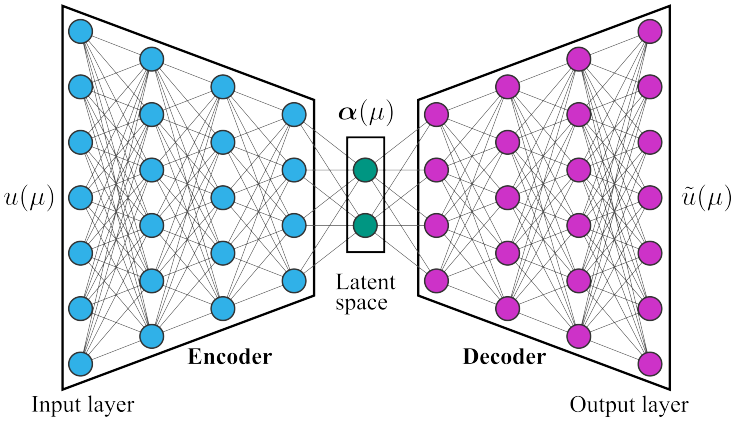


Figure 2.5: A sketch of an autoencoder. High-fidelity snapshots $\mathbf{u}(\mu)$ are compressed by the *Encoder* into a low-dimensional *latent space* $\boldsymbol{\alpha}(\mu)$. The *Decoder* is applied to reconstruct fields $\tilde{\mathbf{u}}(\mu)$ from $\boldsymbol{\alpha}(\mu)$. The network is trained to minimize $\|\mathbf{u}(\mu) - \tilde{\mathbf{u}}(\mu)\|$.

The implementation of Autoencoder in the ROM is similar to PODI: a surrogate $\boldsymbol{\alpha}(\mu) \approx \mathcal{Z}(\mu)$ is constructed to map the input parameters to the latent space. Consequently, for any new input μ_{new}, one can obtain the corresponding $\boldsymbol{\alpha}(\mu_{\text{new}})$ [170]. Remind that, for transient problems, recurrent neural networks are more powerful for creating a surrogate model because they consider feedback in sequence [171].

Recently, the *Physical Informed Neural Network* (PINN) architecture has become a focus in scientific machine learning. It enhances the physical generalization of the neural networks [172; 173]. The basic idea can be summarized as follows. Unlike standard networks, physical knowledge is embedded in PINNs by incorporating specialized inputs, outputs, and loss functions.

For instance, in a computational domain, a set of grid points and time instances are used as inputs, and the time-dependent solutions at these points are treated as output targets. The residual of the governing PDEs and boundary/initial constraints is adopted as the loss function. Consequently, one can obtain solutions at any unseen points by simply querying the PINN. Actually, the trained model is expected to generalize better than standard neural networks because it incorporates physical principles.

Some other frameworks can also learn the control PDE directly from data. The *neural operator* was designed for learning functional maps between input and output spaces [174; 175]. Steven L. Brunton et al. have proposed a so-called SINDy platform to discover governing equations from data [176; 177].

2.3.5 ROM for nuclear engineering

As the dissertation aims to address flow dynamics problems in nuclear reactors, a brief overview of the specific applications is presented below. Readers may also refer to the review publications [178; 179; 180; 181].

Note that only analyses directly focused on practices for nuclear reactors are cited hereafter. Papers that treat the application as background while analyzing benchmark cases are not included.

Based on the references, the implementation of ROM in this topic is not limited to high-fidelity CFD. ROMs are also applied to approximate solutions of lumped-parameter methods. Thus, the existing studies regarding the two different FOM frameworks are introduced.

2.3.5.1 Lumped-parameter models

The description will start with intrusive techniques based on the POD-Galerkin methods. Some early studies were published by Dennis P. Prill, Andreas G. Class, et al. The authors have successfully analyzed several transient thermal-hydraulic phenomena in boiling water reactors. This includes the Korteweg-de Vries (KdV) equation (one-dimensional) for surface waves in shallow water and channel flow entering a free surface domain, as well as natural convection in a closed circuit [182; 183; 184; 185; 186]. M. Zarei has created a ROM for 1D coupled thermo-neutronic problems [187]. Note that neutronics is generally formulated as scalar diffusion equations with source terms, the same as those discussed below.

Remember that the lumped-parameter analysis can be applied to multiphysics phenomena, which are well-suited to nuclear engineering. It is flexible and can be incorporated with non-intrusive ROMs. Radaideh performed uncertainty analyses for problems involving neutronics, reactor kinetics, fuel depletion, thermal-hydraulics, and fuel performance [163]. Machine learning techniques, e.g., RBF and GPR, and deep neural networks were employed to construct

surrogates for all the physics models. The dissertation [163] is highly recommended, as it illustrates applications in many aspects.

Two similar studies about non-intrusive ROM for the transient thermal-hydraulics of the entire reactor system were published by Qi Lu et al. [188] and Young Suk Bang et al. [189], respectively. They aim to approximate solutions of 1D system codes and carry out uncertainty analysis. Their ROMs are constructed employing multilayer neural networks.

2.3.5.2 High fidelity models

The incorporation of ROM and CFD results in more possibilities, and model order reduction is achieved employing distinct procedures.

Alberto Sartori et al. [162] create a POD-Galerkin ROM for thermo-neutronics problems in a single channel in a liquid lead-cooled reactor core. It is worth highlighting the contribution of Péter German et al. [190]. They have proposed a specific multi-physics solver, *GeN-Foam*, for reactor analysis based on OpenFOAM [191]. Furthermore, they constructed intrusive ROMs that included parametric fluid dynamics, heat transfer, and neutron diffusion for molten salt reactors [192; 190; 193]. Rabab Elzohery et al. [194; 195; 196] also contribute to this field by proposing POD-Galerkin ROMs for transient neutronic phenomena.

Recently, non-intrusive techniques have become visible in the observed reference. Dimensionality reduction (e.g., POD, dynamic mode decomposition, active space, etc) with interpolation is applied to several cases: (i) Navier-Stokes flow for a simplified molten salt reactor system [164]; (ii) thermo-neutronics for several benchmark cases [197]; Neutron physics in an entire reactor core [198].

In addition to machine learning algorithms, neural networks are practical to create surrogates between parameters and low-dimensional representations. Huilun Kang et al. [199] conducted boundary condition parametrization for flow in two subchannels. They employed POD and deep neural networks for uncertainty analysis of transient neutronic behavior, as seen in [200; 201; 202; 203]. The integration of data-driven frameworks is also applicable to multiphysics problems in nuclear engineering. Hanxing Liu et al. [204] managed to couple thermal-hydraulics and neutronics. Junda Zhang et al. [205] solved a multiphysics problem containing neutronics, thermal-hydraulics, and structural mechanics.

In addition to the global ROM strategy mentioned above, local ROM techniques can also be adopted in nuclear engineering. Stefano Riva et al. [206] investigated a case that contains different domains for neutronics and unsteady heat conduction. They created two PODI-based ROMs for each region and proposed an iterative framework for coupling them.

Data-driven methods can be effective in solving inverse problems. A series of studies by Stefano Riva et al. successfully employed non-intrusive techniques to optimally select sensor locations for experiments. Additionally, they utilized data assimilation methods to reconstruct the entire

physical field using limited measurement data. The references are listed here [207; 208; 209; 210; 211; 206].

The data-driven framework can also be incorporated with traditional techniques. Mauricio E. Tano et al. [212] improved RANS turbulence models employing Bayesian optimization algorithms and DNS data. Then, they used the modified thermal-hydraulic model to simulate a molten salt-cooled reactor.

Despite building a ROM, the reduction of computational cost can be achieved in a different manner. Huilun Kang et al. exploited neural networks to map CFD fields from coarse and fine meshes [213]. The framework aims to perform CFD simulations on a coarse mesh and reconstruct high-resolution fields on a fine mesh using an ANN.

2.4 Local reduced order models

As described in Section 1.4, symmetry and periodicity are common for industrial devices. Given that similar physical phenomena occur within these repeating structures, it is reasonable to employ local ROM techniques for these applications.

Two comprehensive reviews have been published on this topic: one focusing on intrusive [214] algorithms and the other covering non-intrusive [215] techniques. Nevertheless, this topic is currently under active development, with many advanced methods having been proposed in the past five years. These state-of-the-art studies are included in a recent review by the author [2].

Now, let us briefly summarize the general methodology of local ROMs. Similar to conventional model order reduction techniques, it also consists of the offline and online stages. The main difference from the global framework lies in the domain decomposition and coupling. Each division constructs its own local ROM independently. During the online phase, these local approximations are coupled to obtain the global solution.

“Compared to global ROMs, the local approach offers several significant advantages [216; 217; 89; 218; 219]:

- Dominant features are extracted locally, leading to a better approximation of dominant variations in each subdomain.
- Computational costs decrease for high-fidelity simulations, dimensionality reduction, and the construction 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.”[5]

[5] rephrased from Ruan et al. [2], page 4, 2nd paragraph.

Due to the special procedures of local ROMs, their construction is more complex than that of global ROMs. In addition to the steps mentioned above for global ROMs, two general aspects need to be considered: (i) partition strategies, (ii) local dimensionality reduction procedures. Then, various algorithms have been proposed to assemble subdomain-level reduced systems. They are categorized and summarized hierarchically.

The classification is organized as in Fig. 2.6. Two preprocessing steps are first considered, i.e., domain decomposition and local reduced space generation. Then, the intrusive and non-intrusive techniques for assembling local approximations are discussed. The sub-items in each category will be clarified in the following sections.

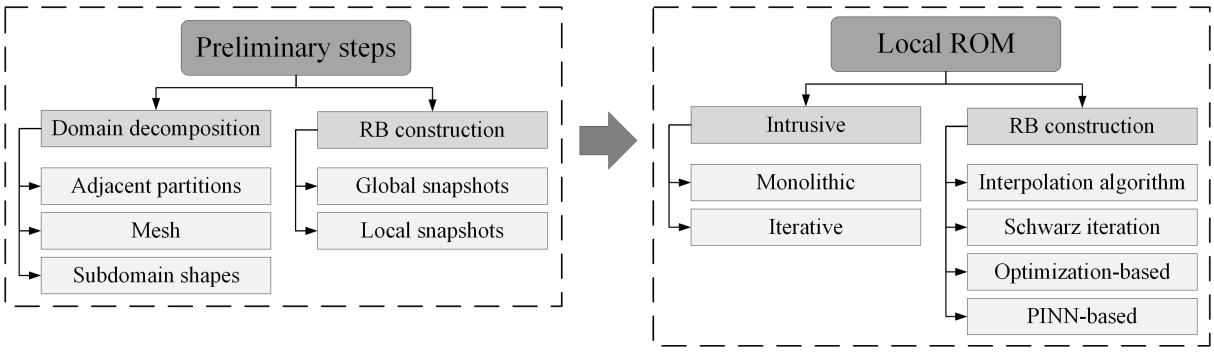


Figure 2.6: Hierarchy and classification of local ROM.

2.4.1 Preliminaries

2.4.1.1 Domain decomposition

As mentioned before, three aspects should be considered for domain decomposition, i.e., overlapping or non-overlapping partitions, mesh conformity at interfaces, individual or generic subdomains, as illustrated in Fig. 2.7. Their features are summarized as follows:

- Subdomains can be overlapping or non-overlapping with their neighbors. The shared regions can enhance the stability and convergence of coupling techniques [45]. However, overlaps introduce challenges in partitioning. The non-overlapping configuration offers greater flexibility and adaptability in dividing the domain, especially for complex geometries.
- Meshes might be conformal or non-conformal at the interfaces among adjacent divisions. The non-conformal mesh is more flexible for FOM simulations, but additional mutual interpolation procedures are required in coupling [220].
- The global model can be decomposed into distinct partitions. In contrast, for geometries with repeating patterns, one or a few reference subdomains can be defined, and the entire

domain is assembled by instantiating and geometrically transforming these generic shapes, as presented in [216; 221]

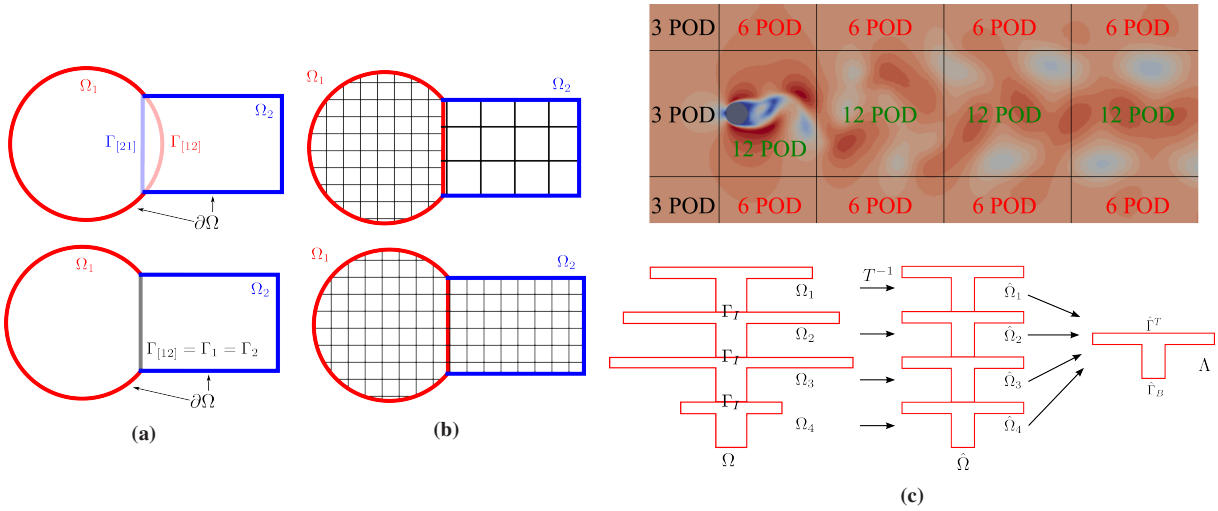


Figure 2.7: Three aspects of domain decomposition: (a) overlapping and non-overlapping partitions; (b) conforming and non-conforming mesh at interfaces; (c) individual (upper) and generic (bottom) subdomains, redrawn based on [152; 38].

2.4.1.2 Snapshots and dimensionality reduction

Generating snapshots is a prerequisite for constructing ROMs. Among existing studies, FOM simulations can be performed on the entire domain or several generic subdomains. The following paragraphs will discuss three scenarios: two adopt global solutions, and one incorporates local results.

Global snapshots and individual decomposition "Let us discuss the first condition, in which FOM simulations are carried out globally, and divisions have distinct shapes. The reduced basis can be computed in two ways.

- Reduced basis functions for the global snapshots are computed. Then, the global modes are divided based on the domain decomposition. That is known as *localized global RB*, see [133; 222; 218]. Note that, in this case, the basis vectors are not orthogonal in each subdomain.
- The global snapshots are first partitioned. Then, dimensionality reduction is applied to each local snapshot set. The subdomain modes are orthogonal and optimal for representing the specific local fields. The implementations can be found in [223; 224; 76; 119] and [1]."[6]

[6] rephrased from Ruan et al. [2], page 6, 5nd paragraph.

The second case considers global solutions, and several generic subdomains assemble the domain. It is achieved by two steps: (i) subdomain solutions are extracted from the global results; (ii) the local snapshots for each generic subdomain are stacked together, and dimensionality reduction is applied to compute the local basis functions. So, only a few sets of basis vectors are generated and then instantiated to represent the entire domain. See [219; 225] and [1] for examples.

Localized training and oversampling The third strategy is called *Localized training*. It assumes that the total domain is constructed from reference blocks. FOM simulations are only performed on these generic subdomains, and reduced bases are computed accordingly.

In this manner, to capture the influence of adjacent regions, the boundary conditions of the reference partitions should be parameterized to account for all possible behaviors across the entire domain. Interested readers can refer to [226; 227; 228].

However, the localized training procedure may not well capture the interactions among neighboring divisions. Thus, a so-called *oversampling* technique is proposed to enhance the representation ability of the localized training methodology. The key modification of the idea is to carry out FOM simulations in a slightly larger domain that includes the generic subdomain. The collection of snapshots and the computation of local basis functions are then performed. The implementations of oversampling can be found in [229; 230].

Once the local RBs are obtained, the next step is to assemble them. In the following sections, both intrusive and non-intrusive frameworks will be discussed.

2.4.2 Intrusive frameworks

Similar to the global ROMs, the intrusive local ROM is typically built upon the Galerkin projection. It can be classified into two categories, i.e., monolithic and iterative procedures. The former assembles all the local sub-problems into a whole system. The latter consists of a set of subdomain-level reduced systems, which are solved separately and iteratively exchange information with neighbors until convergence. Fig. 2.8 summarizes the categorization and existing methods for each type.

2.4.2.1 Monolithic formulations

Reduced Basis Element Method and variants Now, let us discuss the observed monolithic group. The following paragraphs will cover only the basic ideas and references for each method.

The first to be mentioned is the *Reduced Basis Element Method* (RBEM) proposed by Yvon Maday and Einar M. Rønquist in 2002 [231]. To the best of available knowledge, the RBEM

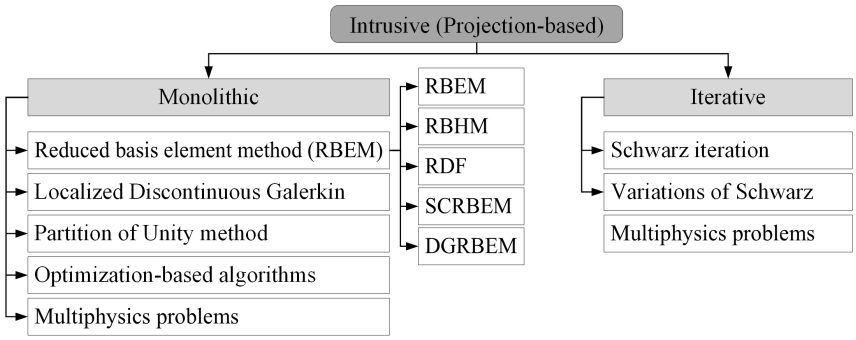


Figure 2.8: Categorization of intrusive techniques. The abbreviation: *Reduced Basis Hybrid Method* (RBHM), *Reduced basis, Domain Decomposition, and Finite element* method (RDF), *Static Condensation RBEM (SCRBEM)*, and *Discontinuous Galerkin RBEM* (DGRBEM).

is the first technique developed to combine domain decomposition and ROMs. It adopts non-overlapping partitions, and reduced basis functions are computed via localized training: FOM simulations are performed only in reference blocks, and the reduced bases are then instantiated to represent the entire domain.

RBEM can be regarded as a combination of finite element and reduced basis methods. Each subdomain is treated as an element transformed from a reference shape. The RBs are also transformed accordingly. As the basis functions are not continuous at the interfaces, Lagrange multipliers are introduced to ensure the continuity of solutions. The methods have been applied to various problems, including heat conduction (Laplacian equation) in a multi-level thermal fin [232], steady Stokes and Navier-Stokes flow in a 2D blood vessel geometry [233; 216]. Two more recent studies have been published for other scenarios, i.e., time-harmonic Maxwell's equation [234] and 3D flow in blood vessels [221].

The ideology of RBEM has been extended to several variants:

- *Reduced Basis Hybrid Method* (RBHM) [235]. It is almost the same as RBEM, except for the basis. In addition to local RBs, three extra functions are enriched as RBs for each partition to improve continuity and consistency at interfaces. Those are: (i) velocity *supremizer* functions [123] to enhance the stability of incompressible flow; (ii) solutions of a coarse mesh simulation for the entire domain and corresponding velocity supremizers. Its application for steady Stokes and Navier-Stokes equations can be found in [235].
- *Reduced basis, Domain Decomposition, and Finite element* (RDF) method [236; 38]. The improvement is also related to the RB. The authors compose the basis with two parts: (i) data-based modes at interiors of subdomains; (ii) finite element basis functions are used to better represent the interface values. Thus, this framework considers additional unknowns at the interfaces. See also in [237].
- *Static Condensation Reduced Basis Element Method* (SCRBEM). This approach was developed by Anthony T. Patera et al. [238; 121; 239; 240; 241; 242; 243]. It includes two RBs for internal and boundary regions, which are called *bubble* and *port* modes,

respectively. The main idea of *static condensation* is to express the unknowns at the interior as a function of those at the interfaces. Thus, even though two sets of basis functions are adopted, the global system contains only degrees of freedom at the boundaries, significantly reducing the final system's size.

- *Discontinuous Galerkin Reduced Basis Element Method* (DGRBEM). It was proposed by Paola F. Antonietti et al. [217] as a generalization and an improvement of RBEM, RBHM, and RDF. It follows the RBEM philosophy. Instead of using Lagrange multipliers, the discontinuous Galerkin (DG) framework is used to derive a coupled system of various subdomains. DG is a combination of finite element and finite volume frameworks that can handle discontinuous basis functions at interfaces. This feature makes DGRBEM particularly well-suited for assembling local modes. See more in [244; 245; 246; 247].

Localized discontinuous Galerkin Now, let us discuss several other approaches based on DG formulations. They are categorized as *localized discontinuous Galerkin* methods for the usage of subdomain-level RBs and DG coupling formulations.

"Andrea Ferrero et al. developed a *domain decomposition POD-DG* algorithm, which is almost the same as the DGRBEM except for spatial division and RBs generation strategy [133; 248; 249]. The approach considers individual decomposition and builds local POD RBs utilizing global solutions. Indeed, it can be regarded as a simplification of DGRBEM. As illustrated in [133], the method results in good accuracy and significant acceleration for the 2-D Euler and RANS equations with the Spalart-Allmaras turbulence closure."[7]

It should be highlighted that the DG-based formulations for FOM and ROM are consistent except for the basis vectors. Interested readers can refer to [222; 218] for the implementation and applications.

Localized Reduced Basis Multiscale (LRBMS) method is another DG-based ROM, which is similar to Ferrero's approach in many aspects (e.g., decomposition and RBs computation). It also incorporates the online RB enrichment technique to balance accuracy and efficiency, as discussed in [127; 250] [8]. It has been used to investigate several problems, such as elliptic equations [125], two-phase flow in porous media [251; 252], and scalar parabolic evolution equations [253].

Partition of Unity method The *Partition of Unity Method* (PUM) [254; 255] was originally developed in the frame of the Generalized Finite Element Method (GFEM). It employs a set of weight functions to couple local bases defined in overlapping regions and then construct a global

[7] rephrased from Ruan et al. [2], page 29, 7th paragraph.

[8] This strategy dynamically extends the reduced basis during the online phase. Instead of relying solely on a precomputed RB, error estimators are utilized to identify poorly approximated regions. New basis functions are then computed and added to the reduced space adaptively, enhancing accuracy when needed.

basis. The weight functions are specifically chosen to ensure the continuity of the global basis in the entire model. In short, the procedure consists of three steps: (i) compute local modes; (ii) assemble them via the partition of unity functions into a globally continuous basis; (iii) solve the problem as a standard global ROM.

Studies regarding the method include: second-order elliptic PDEs [256; 214; 257], time-dependent parabolic equations [258], and Navier-Stokes flow around cylinders [259]. In [229], the localized training with the oversampling technique is also incorporated.

There is a so-called *Multiscale Finite Element Method* (MsFEM) that shares similarities with the aforementioned PUM-based ROM. It couples all finite element basis functions in each partition as local basis vectors. Since finite element functions are initially continuous at interfaces, PUM is not required in this framework. It was applied to several problems, e.g., to linear and nonlinear elliptic problems, [260; 261; 262].

The integration of MsFEM-based ROM with localized training and oversampling techniques is presented in [263; 264; 265; 266; 226], and in addition with adaptive enrichment in [267].

Optimization-based algorithms It should be noted that the main objective of all coupling techniques is to ensure the consistency of local solutions at interfaces. Optimization-based methods formulate the coupling as a minimization problem, subject to constraints of PDE residuals, interface discontinuities, and boundary and initial conditions. Thus, the focus is on how to define the objective function properly and efficiently solve the optimality problem.

Ivan Prusak et al. applied the methods for two incompressible Navier-Stokes benchmarks: the stationary backward-facing step and lid-driven cavity flow [268; 119; 269]. A follow-up study by Tommaso Taddei et al. [230] analyzed flow dynamics (Navier-Stokes equations) in blood vessel systems, and also involved localized training with oversampling. Additionally, in [225; 219], the *static condensation* is utilized to reduce the unknowns of the ROMs.

"Remark that, except for the standard Galerkin approach indicated above, the *Least Squares Galerkin* (LSG) formulation is also employed to build local ROMs. A LSG system is formulated as an *optimal control* problem, aiming to minimize the residual in a least-squares sense."[9] Consequently, the interface constraints can be naturally extended. See [122] for its application to parameterized Laplacian and Burgers' equations in a non-overlapping partitioned computational domain

Multiphysics problems The local ROM framework is particularly suitable for multiphysics problems, as different phenomena can be solved in different subdomains. However, the coupling of several physics is not the same as a single continuum. We recommend [88] and [93] for a general description and applications about the topic. A separate overview is addressed here.

[9] rephrased from Ruan et al. [2], page 34, 2nd paragraph.

Among multiphysics, Fluid-Structure Interaction (FSI) is especially relevant in many practical scenarios. Monica Nonino et al. [224; 270; 271; 272] have published a series of papers on local ROM for the FSI model. They adopted *Lagrange multipliers* to weakly enforce the interface conditions. Similar techniques and applications can be found in [273; 76]. Readers can turn to [274; 275] for modeling aeroelasticity (i.e., Euler equations and linear/nonlinear elastic structure) and [275; 276; 277] for simulating the cardiovascular system (i.e., incompressible Navier-Stokes flow and linear elasticity). Also, studies of other multiphysics problems are available, e.g., nonlinear elastic-plastic structural dynamics and the electro-mechanical system [278; 279; 280].

Lastly, let us mention two exceptions in constructing FSI-ROMs. Although a FSI problem usually involves two physics, the two fields can be considered as a whole. Since the coupling conditions at the fluid-structure interface ensure continuity of certain variables, such as displacement and velocity. A global RB can be extracted from the combined snapshots of both fluid and structure domains. Then, it leads to a standard global ROM, as in [281; 223].

Besides, in some scenarios, the interaction between two physics is one-way. That means one sub-problem provides input to the other without feedback. Thus, the difficulty of interface discontinuity and coupling is avoided. See such examples in [220; 282; 283; 284].

2.4.2.2 Iterative procedures

The *iterative* schemes are commonly used in high-fidelity numerical analysis. The *Schwarz* method and its variations have been widely adopted for domain decomposition problems [285; 45]. Instead of deriving a monolithic system, the approach decomposes the problem into sub-problems and solves them separately. In each iteration, data are exchanged among adjacent divisions to ensure consistency at interfaces. Thus, each interface is considered as a local boundary of two neighboring subdomains.

The framework is flexible and general. It can couple sub-problems based on any discretization techniques (e.g., finite differences, finite element, finite volume, and spectral element methods) [45]. Since the past decade, it has been adapted to obtain the local ROM for either overlapping or non-overlapping partitions.

Schwarz method The first to be mentioned is the Schwarz scheme, which is developed for overlapping subdomains. It transmitted the equality of variables (i.e., Dirichlet condition) in interfaces. The early stage of the iterative scheme is addressed as a hybrid technique to couple FOMs and ROMs [286; 287; 288].

Recently, the iterative methodology has been extended to local ROM, for instance, [289; 290]. Note that overlapping partitions are typically required for the Schwarz iteration, but a recent publication [290] demonstrates that a stable and accurate coupled model can be obtained for non-overlapping cases.

Variations of Schwarz algorithm For non-overlapping divisions, the *Dirichlet-Neumann* (D-N) and *Robin-Robin* (R-R) conditions should be adopted to ensure that the solutions to the subsystems are also solutions to the original problem (see [285]). The D-N pair prescribes Dirichlet conditions on one side of the interface and Neumann conditions on the other side. The R-R pair applies Robin conditions on both sides.

The Dirichlet-Neumann approach is more popular in the observed references, and it is used in elliptic PDEs [291; 292], nonlinear wave propagation [293], incompressible Navier-Stokes equations [294], and solid mechanics [295; 296]. An innovation in [297] effectively addresses the challenges of coupling non-conforming meshes.

The Robin-Robin approach is presented in [227], which analyzes a series of parametrized nonlinear problems, including advection-diffusion-reaction coupling, time-dependent diffusion equations in multiple domains, etc.

Multiphysics problems The iterative framework is particularly suitable for multiphysics problems, as different phenomena can be solved individually and then communicated via interface conditions. Its implementation for FSI problems is published in [298; 299; 300].

"For other multiphysics problems, different interface constraints should be enforced, but the iterative algorithm remains the same. Those include: (i) linear thermo-poroelasticity [301]; (ii) nuclear reactor thermal-hydraulics (Navier-Stokes flow with energy balance) coupled with neutronics (scalar diffusion equation) [206]."[10]

2.4.3 Non-intrusive frameworks

Machine Learning (ML) techniques are now widely recognized for their potential to enhance the efficiency and accuracy of ROMs. They are also broadly used for constructing local ROMs. A review paper published in 2021 [215] discussed the combination of ML and domain decomposition. However, over the past five years, there has been a significant increase in research focusing on the non-intrusive local ROMs. The following sections will provide an overview of the current advances in this area.

"The classification of non-intrusive approaches is not as straightforward as that of intrusive ones. That is due to the variety of ML algorithms and the flexibility of incorporating different techniques in coupling. Thus, considering their fundamental principles, four categories have arisen to roughly classify them, as shown in Fig. 2.9. Those are: (i) Schwarz-based iteration; (ii) interpolation algorithm; (iii) optimization-based technique; (iv) *Physical Informed Neural Network* (PINN) based methods."[11]

[10] rephrased from Ruan et al. [2], page 41, 7th paragraph.
[11] rephrased from Ruan et al. [2], page 41, 9th paragraph.

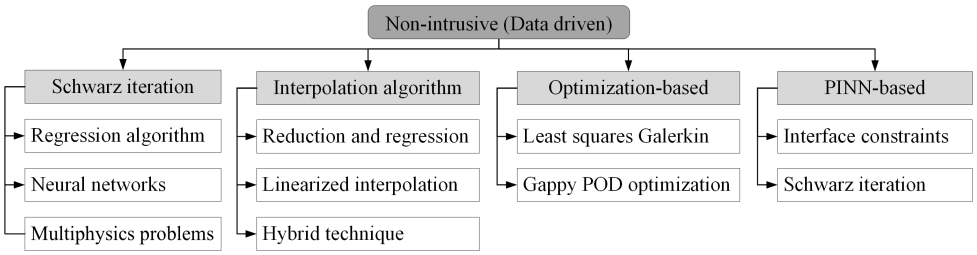


Figure 2.9: Classification of data-driven techniques. The abbreviation: *Physical Informed Neural Network* (PINN).

Note that preliminaries for building data-driven ROMs are already mentioned in Section 2.3.4, where the basic concepts and methodologies of regression algorithms, *Neural Networks*, and *Physics Informed Neural Networks* (PINNs) are briefly introduced.

2.4.3.1 Iterative schemes

The first to be mentioned is the iterative method, which is nearly identical to the iterative intrusive ROMs: a set of local approximation problems is formulated and solved iteratively using interface conditions from neighboring partitions. The major difference is that the local systems are constructed using purely data-driven techniques, without relying on PDEs and Galerkin projections.

See [302] for a study of 2D nonlinear Laplace problems with overlapping subdomains (Dirichlet-Dirichlet). Another investigation published in [228] adopted the Dirichlet-Neumann for non-overlapping conditions (Dirichlet-Neumann). For those two papers, FOM datasets are generated using the localized training strategy, and neural networks are employed to approximate the local solutions.

Iterative approaches are also applied for multiphysics scenarios, such as FSI [303; 304; 305; 306], micro-electro-mechanical-systems [307], and the coupling of neutronics and thermal-hydraulics in nuclear reactor cores [206].

2.4.3.2 Interpolation-based procedures

The so-called interpolation-based procedures also utilize iterations to achieve local approximations. The difference compared to the above schemes is the data exchange between local partitions. Instead of interface quantities, the results from all neighboring partitions are used to surrogate each subsystem.

The three papers, [152; 153; 308], follow the interpolation procedure to tackle 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. In those studies, the authors utilized RBF to create interpolants for local solutions. More complex cases can be seen for [170], where the researchers investigate 2D single-phase flow past a cylinder

and 3D two-phase in-pipe flow. They also compare the performance of various dimensionality reduction methods, including POD, Autoencoder (AE), Adversarial AE, and hybrid SVD-AE.

There are also some variations of the standard interpolation-based procedures, such as the *Subdomain POD-Trajectory-Piecewise-Linearization (TPWL)* algorithm [309; 310; 311], and a hybrid of interpolation and minimization [312; 313].

2.4.3.3 Optimization-based formulations

The optimization can be incorporated in both intrusive and non-intrusive frameworks. The former is discussed in 2.4.2.1. Here, two pure data-driven investigations are presented.

[314; 315] propose an approach that creates an optimality system to minimize the residual of governing PDEs with interface jumps as constraints. In these two papers, the Autoencoder is utilized to extract low-dimensional representations. This nonlinear constrained minimization problem is solved via a so-called *Sequential Quadratic Programming* (SQP). The formulations and implementations of SQP are beyond the scope of this review, and interested readers can find more information in [314; 315].

[316; 317] applied the *Gappy POD* algorithm to couple sub-systems. This technique consists of three steps: (i) local modes are used to approximate each subdomain; (ii) the adjacent regions among partitions are masked as unknowns; (iii) the global modes are used to fill in the masked regions through optimization.

2.4.3.4 Physics-informed neural network (PINN)-based techniques

Now, the focus shifts to physics-informed neural networks (PINNs), which have gained popularity in recent years in scientific machine learning. Descriptions of PINNs can be found in Section 2.3.4.

In terms of local PINN-based ROMs, two procedures have been observed in the literature:

- The coupling is implemented similarly to optimization-based methods. Additional terms are included in the loss function to enforce the interface constraints. Each subdomain is approximated by a separate network, and the model is trained to minimize a global loss function that accounts for all partitions. The approach has been applied to various PDEs, including the 1D Burgers' equation, the 2D steady-state incompressible Navier-Stokes equations, the Korteweg-De Vries equation, and the compressible Euler equations. See [318; 319; 320; 321; 322; 323; 324] for the applications.
- The second strategy is to incorporate the Schwarz iterative framework. An individual network is created for each division. In each iteration, the quantities at interfaces from

neighboring partitions are used as boundary conditions to compute the local loss functions. All sub-networks are solved separately and iteratively exchange information until convergence. Examples can be found in [325; 326; 327; 328; 329; 330; 331; 173; 39].

"Lastly, it should be remarked that PINNs have also become popular for solving multiphysics problems. However, in the papers observed, monolithic networks are constructed to predict physical dynamics, which can not be completely considered as local ROMs. As a hint, references for different applications are listed here: (i) FSI problems [332; 333]; (ii) multiphysics in chemical engineering, including Navier-Stokes, energy conservation, mass transport, and chemical kinematic equations [334; 335]; (iii) electro-thermal coupling (two Laplace equations) [336]."[12]

2.4.4 Discussions

The above literature review includes three parts:

- Conventional numerical tools for nuclear engineering, especially Computational Fluid Dynamics;
- Two groups of reduced order techniques, namely, intrusive (projection-based reduced basis method) and non-intrusive (Proper Orthogonal Decomposition with interpolation);
- The local reduced order model that combines domain decomposition and model order reduction.

The state of the art of the three topics is concluded as follows.

Conventional numerical tools The observed references indicate that CFD has become popular in the reactor industry. It can model/resolve turbulence, revealing sophisticated local flow behaviors in nuclear reactors. Reynolds-Averaged Navier-Stokes, Large Eddy simulation, and Direct Numerical Simulation are three major approaches exploited to model turbulent flow, and they have been widely used for numerical analysis of fluid dynamics and heat transfer in rod bundles.

Regarding the wide range of geometric length scales in the bundle and the limitation of computational resources, RANS remains practical for both academic and engineering research. Thus, RANS is adopted in this dissertation for performing high-fidelity computations.

Reduced order model ROMs denote a type of data-driven methodology for solving physical problems. For its applications in fluid dynamics, it relies on high-fidelity solutions to efficiently perform parametric analysis.

[12] rephrased from Ruan et al. [2], page 58, 4th paragraph.

Before discussing ROMs, three pre-procedures for parametrization are briefly introduced: sampling strategies, geometric parametrization approaches, and parameter space reduction techniques. Then, two major categories, intrusive and non-intrusive methods, are reviewed.

The projection-based reduced basis method (intrusive) became mature in the early 2000s. It involves manipulating governing equations to derive a reduced system. It minimizes the unknowns of the high-fidelity system by several orders of magnitude. This framework has been widely used across various fields and problems, including Poisson equations, Burgers' equations, the Navier-Stokes equations, etc. However, its implementation is generally limited in the academic community, and the applications for industrial-level problems are rare.

The non-intrusive technique has recently become attractive, accompanied by the development of machine learning (and artificial intelligence) techniques. In this research, POD with interpolation is employed to build ROMs. It extracts modes from high-fidelity data and then utilizes regression algorithms to approximate the POD coefficients. The framework is flexible and general for different problems, as it does not require any manipulation of the governing equations. However, its accuracy and stability are not always guaranteed, especially for complex physical scenarios. Sensitivity analysis of hyperparameters is necessary.

In this dissertation, both intrusive and non-intrusive ROMs are investigated and applied to fluid dynamics for typical rod bundle geometries in liquid metal-cooled reactors (see Chapter 4). The accuracy and performance of the two frameworks are compared and discussed for engineering applications.

Local reduced order model The local reduced order model is a novel methodology that combines domain decomposition and model order reduction. It divides the entire computational domain into many subdomains and builds local ROMs in each partition. Then, those local models are assembled to form a global system.

Note that industrial geometries usually contain repetitive patterns, and hence, the application of local ROMs can significantly reduce the cost of generating high-fidelity data. More precisely, in the offline stage, computationally intensive FOM simulations are performed only on a few reference blocks rather than the entire domain, thereby significantly decreasing online costs. Then, reduced bases at the subdomain level are extracted from those local datasets. In the online phase, the local ROMs are instantiated and glued to achieve the global approximation. The characteristics make local ROMs particularly suitable for simulating rod bundles.

Section 2.4 provides a comprehensive review of existing local ROM techniques. The intrusive and non-intrusive frameworks are discussed and categorized. The classification is based on the fundamental principles of coupling local sub-problems. The intrusive methods can be divided into monolithic and iterative procedures. The non-intrusive techniques are classified into four categories: Schwarz-based iteration, interpolation algorithm, optimization-based technique, and PINN-based methods.

Considering the feasibility and practicability for coupling various subdomains, the iterative schemes are adopted (indeed, Dirichlet-Neumann iteration). The PODI is selected to avoid the complexity of equation manipulation and intrusiveness. In short, the *Dirichlet-Neumann iteration-based PODI local reduced order model* is employed. The descriptions and formulation are presented in Chapter 5.

3 Methodology and theory

This chapter presents the methods employed in the research, including the theories, formulations, and implementation details.

For CFD, the Reynolds-averaged Navier-Stokes equations and the OpenFOAM solver are briefly explained. Then, the procedure for high-fidelity solution collection and data reduction is clarified. The most essential parts are the construction of ROMs: the mathematical treatments for deriving the intrusive and non-intrusive reduced systems. The last section specifies the algorithms for achieving local ROMs, along with the domain decomposition strategy and the dimensionality reduction procedure.

3.1 High fidelity computational fluid dynamics

3.1.1 Reynolds-averaged Navier-Stokes equations and turbulence modeling

In the normal operation of LMFRs, the flow is in the turbulent region. *Reynolds-averaged Navier-Stokes equations* are adopted to simulate the dynamics and generate high-fidelity solutions. To simplify turbulence modeling, RANS uses empirical turbulence models with near-wall treatments.

3.1.1.1 Reynolds-averaged Navier-Stokes equations

The derivation of RANS has been addressed in numerous books [23; 24; 337], and therefore, it will be briefly presented here. See also in Appendix A.1.2.1.

Firstly, the incompressible Navier-Stokes equations are defined as

$$\begin{aligned} \frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u}\cdot\nabla)\mathbf{u} &= -\nabla p + \nabla\cdot\left[\nu\left(\nabla\mathbf{u} + \nabla\mathbf{u}^{\mathrm{T}}\right)\right] + \mathbf{f} \\ \nabla\cdot\mathbf{u} &= 0, \end{aligned} \tag{3.1}$$

where the variables denote, $\mathbf{u}$ velocity vector, t time, $p = P/\rho$ kinematic pressure, ρ density, ν kinematic viscosity, $\mathbf{f}$ volumetric source term. The two are the momentum and continuity equations. Note that if $\nu = \text{const}$, $\nabla\cdot\left[\nu\left(\nabla\mathbf{u} + \nabla\mathbf{u}^{\mathrm{T}}\right)\right]$ becomes $\nu\nabla\cdot(\nabla\mathbf{u}) = \nu\nabla^2\mathbf{u}$.

Secondly, by employing *Reynolds decomposition* [338], velocity is written as

$$\mathbf{u} = \overline{\mathbf{u}} + \mathbf{u}' \text{ and } \overline{\mathbf{u}'} = 0,$$

where $\overline{\mathbf{u}}$ represents the time–averaged parts and $\overline{\mathbf{u}'}$ represents velocity fluctuations. The treatment is also adopted for p and $\mathbf{f}$.

The time–averaging operation is applied to equation (3.1) and yields

$$\begin{aligned} \frac{\partial \overline{\mathbf{u}}}{\partial t} + (\overline{\mathbf{u}} \cdot \nabla)\overline{\mathbf{u}} &= -\nabla \overline{p} + \nabla \cdot \left[\nu \left(\nabla \overline{\mathbf{u}} + \nabla \overline{\mathbf{u}}^{\mathrm{T}}\right)\right] + \overline{\mathbf{f}} - \nabla \cdot (\overline{\mathbf{u}'\mathbf{u}'}) \\ \nabla \cdot \overline{\mathbf{u}} &= 0. \end{aligned} \tag{3.2}$$

The nonlinear term $-\nabla \cdot (\overline{\mathbf{u}'\mathbf{u}'})$ represents contribution of velocity fluctuation. Following the *Boussinesq hypothesis* [339], $-\overline{\mathbf{u}'\mathbf{u}'} = \nu_t \left(\nabla \overline{\mathbf{u}} + \nabla \overline{\mathbf{u}}^{\mathrm{T}}\right) - \frac{2}{3} k \mathbf{I}$. Here, ν_t is the turbulence (or eddy) viscosity, which is computed through turbulence models. $k = \frac{1}{2}\overline{\mathbf{u}' \cdot \mathbf{u}'}$ is the turbulence kinetic energy, and $\mathbf{I}$ is the identity matrix. The second term on the right-hand side is usually absorbed into the pressure term by defining a modified pressure $\overline{p}^* = \overline{p} + 2/3k$. But in the following text, the original pressure notation $\overline{p}$ is still used for simplicity.

Consequently, equation (3.2) can be rewritten as

$$\begin{aligned} \frac{\partial \overline{\mathbf{u}}}{\partial t} + (\overline{\mathbf{u}} \cdot \nabla)\overline{\mathbf{u}} &= -\nabla \overline{p} + \nabla \cdot \left[(\nu + \nu_t) \left(\nabla \overline{\mathbf{u}} + \nabla \overline{\mathbf{u}}^{\mathrm{T}}\right)\right] + \overline{\mathbf{f}} \\ \nabla \cdot \overline{\mathbf{u}} &= 0. \end{aligned} \tag{3.3}$$

In this research, the steady-state RANS without a source term is adopted, that is

$$\begin{aligned} (\overline{\mathbf{u}} \cdot \nabla)\overline{\mathbf{u}} &= -\nabla \overline{p} + \nabla \cdot \left[(\nu + \nu_t) \left(\nabla \overline{\mathbf{u}} + \nabla \overline{\mathbf{u}}^{\mathrm{T}}\right)\right] \\ \nabla \cdot \overline{\mathbf{u}} &= 0. \end{aligned} \tag{3.4}$$

It can be concluded from the above derivation that the key idea of RANS for approximating turbulence is to compute ν_t, which is accomplished by the turbulence model described below.

3.1.1.2 Turbulence models and near wall treatments

Many turbulence models have been proposed to compute the turbulence viscosity, such as *Prandtl's mixing length* [340], *Spalart–Allmaras* [341], $k-\epsilon$ [51; 342] and its variations, *Wilcox* $k-\omega$ [343], *Menter Shear–Stress Transport* (SST) $k-\omega$ [344], *Reynolds stress equation* [345], etc. Their formulations are presented in [23].

SST $k-\omega$ model Note that the standard $k-\epsilon$ model is suitable for high-Reynolds-number flows but cannot well represent the boundary layer region at low Reynolds numbers. The *Wilcox* $k-\omega$ can better capture the near-wall region while requiring a much finer mesh. In this research, the SST $k-\omega$ model is used. As indicated by Florian Menter, this hybrid approach combines the advantages of both models. It applies $k-\omega$ in the near-wall area and switches to $k-\epsilon$ in the fully turbulent bulk flow [344; 346; 347].

The SST $k-\omega$ model involves two new unknowns: turbulence kinetic energy k and turbulence-specific dissipation rate ω. The eddy viscosity ν_t is computed as $\nu_t = k/\omega$. Thus, two additional transport equations for k and ω are yielded as closure for the RANS system. That denotes

$$\begin{aligned}
\frac{\partial k}{\partial t} + \mathbf{u}\cdot\nabla k &= \nabla\cdot[(\,\nu+\sigma_k\nu_t\,)\nabla k] + P_k - \beta^* k\omega, \\
\frac{\partial \omega}{\partial t} + \mathbf{u}\cdot\nabla \omega &= \nabla\cdot[(\,\nu+\sigma_\omega\nu_t\,)\nabla \omega] + \frac{\gamma}{\nu_t}P_k - \beta w^2 + 2\,(1-F_1)\,\frac{\sigma_{\omega 2}}{\omega}\nabla k\cdot\nabla\omega,
\end{aligned} \tag{3.5}$$

where $P_k = 2\nu_t \mathbf{S}_{ij}\mathbf{S}_{ij}$ ($\mathbf{S}_{ij} = 1/2(\nabla\overline{\mathbf{u}} + \nabla\overline{\mathbf{u}}^{\mathrm{T}})$), F_1 is a blending function, and $\beta^*, \beta, \sigma_k, \sigma_\omega, \sigma_{\omega 2}$ are empirical constants. This yields

$$\nu_t = \frac{a_1 k}{\max(a_1\omega, S_t F_2)},$$

where a_1 is a constant, $S_t = \sqrt{2\mathbf{S}_{ij} : \mathbf{S}_{ij}}$, and F_2 is another blending function.

Note that the SST $k-\omega$ model involves several empirical constants and blending functions to switch between $k-\omega$ and $k-\epsilon$ models. The values of the constants and the definitions of the blending functions F_1 and F_2 are not detailed here, and interested readers can refer to [23; 24; 348] for details.

Near-wall treatment According to boundary layer theory, the velocity varies significantly within the boundary layer [52]. Therefore, accurate near-wall treatment is essential for reliable RANS simulations [23]. For turbulent flows, the variance of the mean velocity profile near a solid boundary (in the direction normal to the boundary), $\overline{\mathbf{u}}\cdot\mathbf{n}$, can be described by the *law of the wall* [22]. The spatial discretization around the wall depends on the turbulence model and Reynolds number, see Appendix B.1.1 for the description.

3.1.2 Finite Volume method and OpenFOAM

Open Field Operation And Manipulation (OpenFOAM), a popular open-source CFD software, is applied for performing FOM calculations [349; 350; 337; 351]. The book [24] is highly recommended for interested readers.

OpenFOAM is based on the *Finite Volume Method* (FVM) for spatial discretization, which is a widely used approach in CFD. It supports both structured and unstructured meshes. The introduction of FVM and meshing is briefly presented in Appendix B.1.2.1 and B.1.2.2, respectively.

Once the mesh is available, the integral PDEs for all cells, along with the boundary conditions, can be assembled into a global system, which is solved iteratively. In this research, the incompressible steady-state solver *simpleFoam* is employed to perform RANS simulations (see Appendix B.1.2.3). The solver is built based on the *Semi-Implicit Method for Pressure-Linked Equations* (SIMPLE) iterative algorithm [352; 353].

Generally, after discretization and linearization, the algebraic systems are solved iteratively. Various techniques can be adopted to solve these linear equations, such as *Gauss-Seidel*, *Preconditioned Bi-conjugate Gradient*, *Geometric Algebraic MultiGrid* methods, etc. [337]. Typically, the solving approaches do not affect the solution accuracy once the problem has converged. Whereas, they might significantly affect the convergence rate and consequently influence computation time.

Now, it is time to introduce the FOM simulation configuration in this research. SimpleFoam with RANS is utilized to generate high-fidelity solutions. Both structured and unstructured meshes are employed, depending on the geometry complexity. To balance the convergence rate and computational cost, the *Preconditioned Biconjugate Gradient* and *Geometric Algebraic MultiGrid* algorithms are used to solve the algebraic systems.

3.2 Reduced order model for fluid dynamics in rod bundles

Both intrusive and non-intrusive techniques are utilized for constructing ROMs. Their fundamental ideologies are already illustrated in Section 2.3. The following subsections will involve the methods adopted in this dissertation.

The descriptions focus on mathematical aspects regarding the methods. Parameter selection and snapshot collection are described as preliminaries. Formulations for obtaining POD modes will be provided. The implementation of the POD-Galerkin ROM for RANS is then addressed. The treatments for boundary conditions and turbulence are briefly introduced. For non-intrusive approaches (i.e., PODI), the mathematical procedures of interpolation techniques are presented. The last part consists of the algorithm to achieve local ROMs, including domain decomposition, dimensionality reduction, and an iterative scheme.

3.2.1 Preliminaries

3.2.1.1 Parameters and snapshots

As indicated before, both physical properties and geometrical shapes can be regarded as parameters. In this research, only the physical variables (i.e., *fluid viscosity* and *Dirichlet boundary condition*) are involved.

Notation for snapshots and reduced basis are already explained in Section 2.3. For completeness, those are recalled here. Considering a parameter space $\mathcal{P}$, a set of samples $\boldsymbol{\mu}_j \in \mathcal{P}, j = 1, \ldots, N_\mu$ is selected. High fidelity solutions $\mathbf{u}(\boldsymbol{\mu}_j)$ corresponding to each $\boldsymbol{\mu}_j$ are collected column-wise as a snapshot matrix $\mathbf{S}$. Note that $\boldsymbol{\mu}_j$ can also be multi-dimensional, i.e., $\boldsymbol{\mu}_j = [\mu_{j,1}, \mu_{j,2}, \ldots, \mu_{j,N_\mathcal{P}}]^{\mathrm{T}} \in \mathbb{R}^{N_\mathcal{P}}$, where $N_\mathcal{P}$ is the dimension of the parameter space.

The dimensions of each $\mathbf{u}(\boldsymbol{\mu}_j) \in \mathbb{R}^{N_h}$ are proportional to the mesh size, which is usually in the order of millions. In general, physical fields can be well approximated by several dominant features [354]. Thus, dimensionality reduction techniques are proposed to identify a few variables α_i as low-dimensional representations of $\mathbf{u}(\boldsymbol{\mu}_j)$. POD has been widely used for this task for decades, and it is adopted in this study.

3.2.1.2 Proper orthogonal decomposition

The POD can achieve the approximation $\mathbf{u}(x; \boldsymbol{\mu}_j) \approx \sum_{i=1}^{N^{\mathrm{RB}}} \alpha_i(\boldsymbol{\mu}_j)\mathbf{v}_i(x)$. That denotes a linear combination of orthonormal POD modes $\mathbf{v}_i(x)$ weighted by $\alpha_i(\boldsymbol{\mu}_j)$.

The implementation of POD is briefly described here. Assume a snapshot matrix $\mathbf{S} = \left[\mathbf{u}_1, \ldots, \mathbf{u}_{N_\mu}\right] \in \mathbb{R}^{N_h \times N_\mu}$, where $\mathbf{u}$ is a physical field and $\mathbf{u} \in \mathbb{R}^{N_h}$. Through POD, one can obtain $\mathbf{S} = \boldsymbol{\Phi}_F \mathbf{A}_F^{\mathrm{T}}$, where $\boldsymbol{\Phi}_F \in \mathbb{R}^{N_h \times N_\mu}$ is the full basis matrix and $\mathbf{A}_F^{\mathrm{T}} \in \mathbb{R}^{N_\mu \times N_\mu}$ is coefficient matrix. Each column of Φ denotes a basis function, e.g., $\mathbf{v}_i,\ i = 1, \cdots, N_\mu$. Remember that each $\mathbf{v}_i$ is orthogonal to the others, namely, for $i, j = 1, \ldots, N_\mu$,

$$(\mathbf{v}_i, \mathbf{v}_j) = \begin{cases} 1, & i = j, \\ 0, & i \neq j, \end{cases}$$

where $(\cdot, \cdot)$ is an inner product operator.

The algebraic procedure of POD is sketched in Fig.3.1. Generally, the first several modes can capture most essential flow variances, which denotes the approximation $\mathbf{S}^{\mathrm{RB}} \approx \boldsymbol{\Phi}\mathbf{A}^{\mathrm{T}}$, where $\boldsymbol{\Phi} \in \mathbb{R}^{N_h \times N^{\mathrm{RB}}}$ and $\mathbf{A}^{\mathrm{T}} \in \mathbb{R}^{N^{\mathrm{RB}} \times N^\mu}$, and $N^{\mathrm{RB}} \ll N_\mu$. Suppose $\boldsymbol{\Phi} = [\mathbf{v}_1 | \cdots | \mathbf{v}_{N^{\mathrm{RB}}}]$, the POD reduced basis is spanned by column vectors of $\boldsymbol{\Phi}$, indeed, $\mathcal{V} \coloneqq \mathrm{span}\{\mathbf{v}_1, \cdots, \mathbf{v}_{N^{\mathrm{RB}}}\}$.

The computation of POD modes is not unique. It can be obtained using two equivalent algorithms: the Singular Value Decomposition (SVD) of the snapshot matrix and the eigendecomposition of

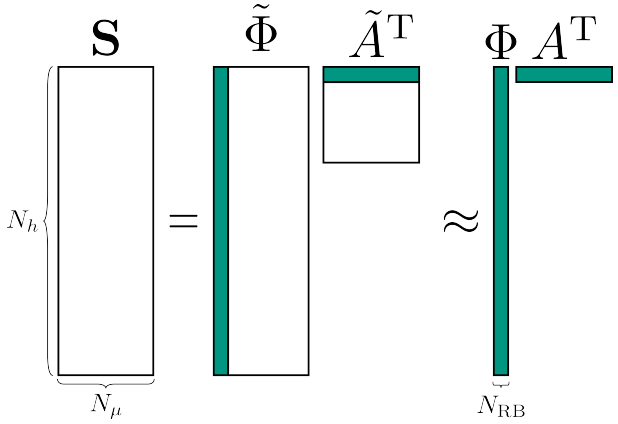


Figure 3.1: Proper Orthogonal Decomposition of a snapshot matrix $\mathbf{S} = \mathbf{\Phi}_F \mathbf{A}_F^{\mathrm{T}}$. Full basis matrix $\mathbf{\Phi}_F$ and corresponding coefficients' matrix $\mathbf{A}_F^{\mathrm{T}}$. N^{RB} basis functions are retained for low-rank approximation, i.e., $\mathbf{S} \approx \mathbf{S}^{\mathrm{RB}} = \mathbf{\Phi}\mathbf{A}^{\mathrm{T}}$. Resolution of snapshots N_h; number of parameter N_μ; RB dimension N^{RB}.

the covariance matrix. Their formulations and procedures are illustrated in Appendix B.2. See also in [33; 30; 94; 89].

The POD approximation error denotes the difference between the original snapshot matrix and its low-rank approximation. It is quantified by the Frobenius norm of the relative error (also known as relative *projection error* or *POD truncation error*), which is given by

$$\mathcal{E}_{N^{\mathrm{RB}}} = \frac{\left\|\mathbf{S} - \mathbf{S}^{\mathrm{RB}}\right\|_F}{\|\mathbf{S}\|_F} = \frac{\left\|\mathbf{S} - \mathbf{\Phi}\mathbf{\Phi}^{\mathrm{T}}\mathbf{S}\right\|_F}{\|\mathbf{S}\|_F}. \tag{3.6}$$

In many realistic applications, $\mathcal{E}_{N^{\mathrm{RB}}}$ decays exponentially [89]. That denotes that the first several POD modes can well represent $\mathbf{S}$. Besides, the so-called *energy norm* is also commonly used as a criterion to quantify the retained POD modes:

$$\mathcal{E}^2_{N^{\mathrm{RB}}} = \frac{\left\|\mathbf{S} - \mathbf{S}^{\mathrm{RB}}\right\|_F^2}{\|\mathbf{S}\|_F^2}, \tag{3.7}$$

which represents the proportion of the energy of the snapshots neglected by the first N^{RB} basis functions [30].

According to the *Schmidt-Eckart-Young* theorem [355; 356], the *best N^{RB}-rank approximation* of a matrix with respect to *Frobenius norm* is its first N^{RB} POD basis functions. Thus, truncation error is regarded as a standard measure of the quality of the ROM.

3.2.2 POD Galerkin method

The mathematical and numerical foundations of Galerkin projection, indeed, strong and weak forms, have been discussed in Section 2.3.3.2. The following sections will focus on the construction of ROMs for steady-state RANS without a source term (see equation 3.4).

Suppose POD is applied to compute the reduced basis, and the basis functions for velocity $\mathbf{u}$ and pressure p are defined as

$$\begin{aligned} \mathbf{u}(\mathbf{x};\mu_j) &\approx \sum_{i=1}^{N_{\mathbf{u}}^{\mathrm{RB}}} \alpha_i(\mu_j)\mathbf{v}_i(\mathbf{x}) \\ p(\mathbf{x};\mu_j) &\approx \sum_{i=1}^{N_p^{\mathrm{RB}}} \beta_i(\mu_j) q_i(\mathbf{x}), \end{aligned} \tag{3.8}$$

where $\mathbf{v}_i$ and q_i are POD modes and α_i and β_i are corresponding coefficients.

By applying Galerkin projection to the steady-state RANS-FV formulations (B.2) in a domain Ω, it results in

$$\begin{aligned} -\int_\Omega \nabla \cdot \left[\nu_{\text{eff}}\left(\nabla\mathbf{u} + \nabla\mathbf{u}^{\mathrm{T}}\right)\right] \cdot \mathbf{v}_i + \int_\Omega (\mathbf{u}\cdot\nabla)\mathbf{u}\cdot\mathbf{v}_i + \int_\Omega \nabla p \cdot \mathbf{v}_i = 0, \\ \int_\Omega \nabla\cdot\mathbf{u}\, q_i = 0, \end{aligned} \tag{3.9}$$

where $\nu_{\text{eff}} = \nu + \nu_t$ is the effective viscosity.

Note that, to derive the final ROM formulations, several essential items should be discussed: (i) treatments for boundary conditions, (ii) turbulence, and (iii) stabilization techniques. These will be introduced as follows.

3.2.2.1 Implementation Aspects

Treatments of boundary conditions In fluid dynamics problems, Dirichlet, Neumann, and Robin are three common boundary conditions. Only Dirichlet and Neumann conditions, defined as Γ_D and Γ_N, are used for this research. Remember that the weak formulation (3.9) does not explicitly contain boundary conditions. Specific approaches are required to impose these boundaries.

There are two common approaches to deal with the Dirichlet boundary in the frame of ROM: (i) the *lifting function* and (ii) the *penalty* method [118; 113].

The lifting function $\mathbf{u}^L$ denotes a specific field defined on the computational domain Ω that satisfies the Dirichlet boundary conditions. Given the velocity in a Γ_D is $\mathbf{u}_D$, $\mathbf{u}^L\big|_{\Gamma_D} = \mathbf{u}_D$.

Now, assume $\mathbf{u}_D$ is independent of the parameter μ_j, velocity can be decomposed into two parts as $\mathbf{u}(\mu_j) = \mathbf{u}^L + \mathbf{u}_o(\mu_j)$, in which $\mathbf{u}_o|_{\Gamma_D} = 0$. Then, POD is applied to the homogenized fields and leads to $\mathbf{u}_o = \sum_{i=1}^{N_\mathbf{u}^{\text{RB}}} \alpha_i \mathbf{v}_i$. In short,

$$\mathbf{u}(\mathbf{x}; \mu_j) = \mathbf{u}^L(\mathbf{x}) + \sum_{i=1}^{N_\mathbf{u}^{\text{RB}}} \alpha_i(\mu_j)\mathbf{v}_i(\mathbf{x}).$$

Hence, the POD modes satisfy the homogeneous Dirichlet conditions, i.e., $\mathbf{v}_i|_{\Gamma_D} = 0$. Be aware that $\mathbf{u}^L$ can also be time-dependent if necessary.

In case the Dirichlet condition is dependent on parameters, this can be achieved by using a set of lifting functions [357], namely

$$\mathbf{u}^L(\mu_i^L) = \sum_{i=1}^{N^L} \mu_i^L \mathbf{u}_i^L,$$

where the total $\mathbf{u}^L(\mu_i^L)$ is expressed as a linear combination of basis functions $\mathbf{u}_i^L$, and the weighting factors μ_i^L denote parameters.

Remember that the velocity is divergence-free for incompressible flow, so the requirement also applies to the lifting function. Thus, $\mathbf{u}^L$ can be computed via several strategies: (i) the average of snapshots $\mathbf{u}^L = \frac{1}{N_\mu}\mathbf{u}(\mu_j)$ [118]; (ii) the solution of potential flow [358]; (iii) the solution of Stokes equation [243].

Now, let us turn to the penalty method. To enforce $\mathbf{u}_D$ on Γ_D, an extra term is added to the weak formulation,

$$\tau_u \int_{\Gamma_D} (\mathbf{u}_D \cdot \mathbf{v_i} - \sum_{j=1}^{N_\mathbf{u}^{\text{RB}}} \alpha_j \mathbf{v}_j \cdot \mathbf{v}_i),$$

where τ_u is a penalty factor that needs to be tuned via sensitivity analysis [118; 136]. Obviously, the penalty term can account for any Dirichlet profiles in the same manner.

Now, the focus turns to the treatment for Neumann conditions. The penalty method can also be extended in this situation [359]. Given the value $\mathbf{g}_N = \nabla\mathbf{u} \cdot \mathbf{n}|_{\Gamma_N}$, one can formulate

$$\tau_g \int_{\Gamma_N} (\mathbf{g}_N \cdot \mathbf{v_i} - \sum_{j=1}^{N_\mathbf{u}^{\text{RB}}} \alpha_j \nabla\mathbf{v}_j \cdot \mathbf{n} \cdot \mathbf{v}_i), \tag{3.10}$$

where τ_g is the penalty factor for $\mathbf{g}_N$. Note that pressure conditions can be treated similarly.

Additionally, for Neumann conditions of velocity $\mathbf{g}_N$, it can be derived directly from the weak form. Let us reformulate the effective viscosity term by considering the integral by parts as

$$-\int_\Omega \nabla\cdot\left[\nu_{\text{eff}}\left(\nabla\mathbf{u} + \nabla\mathbf{u}^{\text{T}}\right)\right]\cdot\mathbf{v}_i = \int_\Omega \left[\nu_{\text{eff}}\left(\nabla\mathbf{u} + \nabla\mathbf{u}^{\text{T}}\right)\right] : \nabla\mathbf{v}_i - \int_{\partial\Omega} \left[\nu_{\text{eff}}\left(\nabla\mathbf{u} + \nabla\mathbf{u}^{\text{T}}\right)\right]\cdot\mathbf{n}\cdot\mathbf{v}_i.$$

Then, $\mathbf{g}_N$ at $\Gamma_N \in \partial\Omega$ is included in the boundary term

$$-\int_{\Gamma_N} \left[\nu_{\text{eff}} \left(\nabla \mathbf{u} + \nabla \mathbf{u}^{\mathrm{T}}\right)\right] \cdot \mathbf{n} \cdot \mathbf{v}_i = -\int_{\Gamma_N} \nu_{\text{eff}} \nabla \mathbf{u}^{\mathrm{T}} \cdot \mathbf{n} \cdot \mathbf{v}_i - \underbrace{\int_{\Gamma_N} \nu_{\text{eff}}\, \mathbf{g}_N \cdot \mathbf{v}_i}_{\text{Neumann condition}}. \tag{3.11}$$

Turbulence RANS based on SST $k-\omega$ model contains two equations for k and ω. Consequently, the system can be solved by applying Galerkin projection to the extra equations, as presented in [360; 361].

Moreover, Giovanni Stabile et al. [113; 136] have proposed a data-driven method as an alternative, which is adopted here. It contains three key steps.

First, eddy viscosity snapshots are collected during the offline stage. Similarly, ν_t is expressed by POD as

$$\nu_t(x;\mu_j) \approx \sum_{i=1}^{N_{\nu_t}^{\text{RB}}} \xi_i(\mu_j)\eta_i(x),$$

where ξ_i and η_i are POD coefficients and modes, respectively.

Secondly, considering the parameter μ_j, each $\xi_i(\mu_j)$ is approximated by

$$\xi_i(\mu_j) \approx \mathcal{Z}(\mu_j),$$

where $\mathcal{Z}$ is a regression or interpolation algorithm, e.g., radial basis function interpolation [113; 136].

Lastly, during the online phase, $\nu_t(\mu_{\text{new}})$ can be interpolated straightforwardly and incorporated into the momentum equation.

Note that the inputs of $\mathcal{Z}$ are not unique. Since turbulence contribution is mainly due to the fluid flow, ξ_i can also be expressed as

$$\xi_i(\mu_j) \approx \mathcal{Z}\left(\{\boldsymbol{\alpha}(\mu_j), \mu^L\}\right),$$

where $\boldsymbol{\alpha}(\mu_j)$ is POD coefficient of velocity modes and μ^L denotes factors of the lifting functions.

Stabilization Techniques The POD can rewrite each snapshot as a linear combination of modes. Therefore, each POD basis function can be correlated inversely to snapshots, which denotes

$$\mathbf{v}_i = \sum_{k=1}^{N\mu} \varsigma_k \mathbf{u}(\mu_k).$$

Due to the incompressible assumption, the snapshots are divergence-free, and consequently, $\nabla \cdot \mathbf{v}_i = 0$. Then, the continuity equation is satisfied anyway, indeed,

$$\int_\Omega \nabla \cdot \sum_{j=1}^{N_{\mathbf{u}}^{\mathrm{RB}}} \mathbf{v}_i \, q_i = 0.$$

Consequently, the reduced PDE (3.9) is underdetermined and can not be solved algebraically. To deal with the issue, two so-called *stabilization techniques* have been proposed.

One is called *supremizer enrichment* [123; 117], which aims to enrich velocity with basis functions that are not divergence-free. The supremizer problem (see [117] for details) is defined in the computational domain Ω as

$$\begin{aligned} -\nabla \cdot \nabla \mathbf{v}^{\mathrm{sup}} &= \nabla q_i \qquad \text{in } \Omega \\ \mathbf{v}^{\mathrm{sup}} &= 0 \qquad \text{in } \partial\Omega, \end{aligned}$$

where $\mathbf{v}^{\mathrm{sup}}$ is the solution, q_i is the pressure modes. Hence, in total N_p^{RB} supremizer modes are obtained. Then, the velocity basis is enriched as $\mathcal{V} = \mathrm{span}\left\{\mathbf{v}_1, \ldots, \mathbf{v}_{N_u^{\mathrm{RB}}}, \mathbf{v}_1^{\mathrm{sup}}, \ldots, \mathbf{v}_{N_p^{\mathrm{RB}}}^{\mathrm{sup}}\right\}$.

Another technique aims to replace the continuity conservation with a *Pressure Poisson equation* (PPE) [362; 363; 118]. It can be obtained by taking the divergence of the momentum equation and applying the continuity condition, which yields

$$\Delta p = -\nabla \cdot (\mathbf{u} \cdot \nabla)\mathbf{u}. \tag{3.12}$$

Through Galerkin projection with q_i, the weak form becomes

$$\int_\Omega \Delta p \, q_i = -\int_\Omega \nabla \cdot [(\mathbf{u} \cdot \nabla)\mathbf{u}] \, q_i. \tag{3.13}$$

According to the literature review, both treatments have been widely applied to construct ROMs for incompressible flow. It can be concluded from the observed references that the enrichment technique is more popular, e.g., [123; 235; 364; 365; 366; 138; 270; 367; 273; 119; 247]. The second approach can be seen in [363; 118; 113; 136]. A comparison of the two methods shown in [368] indicates that they are comparable in accuracy. The comparison is presented in Section 4.1.2.

3.2.2.2 ROM formulation

Now, let us derive the POD-Galerkin ROM in the FVM framework. As mentioned above, two strategies can be applied to handle boundary conditions. The continuity equation can be

solved using either the supremizer enrichment approach or the PPE. The following descriptions illustrate four reduced algebraic systems considering different treatments.

Lifting function and supremizer enrichment To simplify the explanation, let us assume homogeneous boundaries except for a non-homogeneous Dirichlet condition at the inlet. The velocity and pressure are expressed as

$$\begin{aligned} \mathbf{u} &= \mathbf{u}^L + \sum_{i=1}^{N_{\mathbf{u}}^{\mathrm{RB}}+N_p^{\mathrm{RB}}} \alpha_i \mathbf{v}_i \quad \text{with } \mathbf{v}_i \in \mathcal{V}, \\ p &= \sum_{i=1}^{N_p^{\mathrm{RB}}} \beta_i q_i \qquad\qquad\qquad \text{with } q_i \in \mathcal{Q}, \end{aligned}$$

where the N_p^{RB} supremizer modes are included in $\mathbf{v}_i$, $\mathcal{V}$ and $\mathcal{Q}$ are RBs, indeed, $\mathcal{V} = \text{span}\left\{\mathbf{v}_1, \ldots, \mathbf{v}_{N_u^{\mathrm{RB}}}, \mathbf{v}_1^{\mathrm{sup}}, \ldots, \mathbf{v}_{N_p^{\mathrm{RB}}}^{\mathrm{sup}}\right\}$ and $\mathcal{Q} = \text{span}\left\{q_1, \ldots, q_{N_p^{\mathrm{RB}}}\right\}$.

Now, substituting the POD approximations into equation (3.9) and obtain the following algebraic system,

$$\begin{aligned} -B\boldsymbol{\alpha} + \boldsymbol{\alpha}^{\mathrm{T}} C \boldsymbol{\alpha} + C^{L1}\boldsymbol{\alpha} + \boldsymbol{\xi}^{\mathrm{T}} E \boldsymbol{\alpha} + E^L \boldsymbol{\xi} + K\boldsymbol{\beta} &= B^L - C^L \\ A\boldsymbol{\beta} &= 0, \end{aligned} \tag{3.14}$$

where the matrices are referred to as

$$\begin{aligned} A_{ij} &= (q_i, \nabla \cdot \mathbf{v}_j)_{L^2(\Omega)}, \\ B_{ij} &= \left(\mathbf{v}_i, \nabla \cdot \left[\nu(\nabla \mathbf{v}_j + \nabla \mathbf{v}_j^{\mathrm{T}})\right]\right)_{L^2(\Omega)}, \\ C_{ijk} &= (\mathbf{v}_i, (\mathbf{v}_j \cdot \nabla)\mathbf{v}_k)_{L^2(\Omega)}, \\ C_{ij}^L &= \left(\mathbf{v}_i, (\mathbf{u}^L \cdot \nabla)\mathbf{v}_j + (\mathbf{v}_j \cdot \nabla)\mathbf{u}^L\right)_{L^2(\Omega)}, \\ E_{ijk} &= \left(\mathbf{v}_i, \nabla \cdot \left[\eta_j(\nabla \mathbf{v}_k + \nabla \mathbf{v}_k^{\mathrm{T}})\right]\right)_{L^2(\Omega)}, \\ E_{ij}^L &= \left(\mathbf{v}_i, \nabla \cdot \left[\eta_j(\nabla \mathbf{u}^L + \nabla (\mathbf{u}^L)^{\mathrm{T}})\right]\right)_{L^2(\Omega)}, \\ K_{ij} &= (\mathbf{v}_i, \nabla q_j)_{L^2(\Omega)}, \end{aligned}$$

and the left-hand side terms corresponding to the lifting function are given by

$$\begin{aligned} B_i^L &= \left(\mathbf{v}_i, \nabla \cdot \left[\nu(\nabla \mathbf{u}^L + (\nabla \mathbf{u}^L)^{\mathrm{T}})\right]\right)_{L^2(\Omega)}, \\ C_i^L &= \left(\mathbf{v}_i, (\mathbf{u}^L \cdot \nabla)\mathbf{u}^L\right)_{L^2(\Omega)}, \end{aligned}$$

where $(\cdot, \cdot)_{L^2(\Omega)}$ is an L^2 inner product operator over the domain Ω.

Be aware that, even if $\mathbf{u}^L$ comprises several basis fields $\mathbf{u}_i^L$, the ROM system (3.14) remains the same.

Dirichlet penalty and supremizer basis When the Dirichlet boundary, $\mathbf{u}_D$ at Γ_D, is enforced by the penalty method, no boundary terms are needed, the ROM (3.14) is rewritten as

$$\begin{aligned} -B\boldsymbol{\alpha} + \boldsymbol{\alpha}^{\mathrm{T}} C \boldsymbol{\alpha} + \boldsymbol{\xi}^{\mathrm{T}} E \boldsymbol{\alpha} + K\boldsymbol{\beta} + F\boldsymbol{\alpha} &= F^L \\ A\boldsymbol{\beta} &= 0, \end{aligned} \tag{3.15}$$

where the new penalty matrices F and F^L are defined as

$$\begin{aligned} F_{ij} &= \left(\mathbf{v}_i, \mathbf{v}_j\right)_{L^2(\Omega)}, \\ F_i^L &= \left(\mathbf{v}_i, \mathbf{u}_D\right)_{L^2(\Gamma_D)}. \end{aligned}$$

Lifting function and pressure Poisson equation In case the PPE replaces the continuity equation, the momentum equation remains the same. Now, the reduced subspaces become $\mathcal{V} = \text{span}\left\{\mathbf{v}_1, \ldots, \mathbf{v}_{N_u^{\text{RB}}}\right\}$ and $\mathcal{Q} = \text{span}\left\{q_1, \ldots, q_{N_p^{\text{RB}}}\right\}$. ROM system (3.14) can be reformulated as

$$\begin{aligned} -B\boldsymbol{\alpha} + \boldsymbol{\alpha}^{\mathrm{T}} C \boldsymbol{\alpha} + C^{L1}\boldsymbol{\alpha} + \boldsymbol{\xi}^{\mathrm{T}} E \boldsymbol{\alpha} + E^L\boldsymbol{\xi} + K\boldsymbol{\beta} &= B^L - C^L \\ D\boldsymbol{\beta} + \boldsymbol{\alpha}^{\mathrm{T}} G \boldsymbol{\alpha} &= 0, \end{aligned} \tag{3.16}$$

where the other terms remain the same, and the new matrices in PPE are computed by

$$\begin{aligned} D_{ij} &= \left(q_i, \Delta q_j\right)_{L^2(\Omega)}, \\ G_{ijk} &= \left(q_i, \nabla \cdot [(\mathbf{v}_j \cdot \nabla)\mathbf{v}_k]\right)_{L^2(\Omega)}. \end{aligned}$$

Dirichlet penalty and pressure Poisson equation The momentum part of (3.15) and the continuity conservation of (3.16) are combined to yield the ROM considering penalty and the PPE, which results in

$$\begin{aligned} -B\boldsymbol{\alpha} + \boldsymbol{\alpha}^{\mathrm{T}} C \boldsymbol{\alpha} + \boldsymbol{\xi}^{\mathrm{T}} E \boldsymbol{\alpha} + K\boldsymbol{\beta} + F\boldsymbol{\alpha} &= F^L, \\ D\boldsymbol{\beta} + \boldsymbol{\alpha}^{\mathrm{T}} G \boldsymbol{\alpha} &= 0. \end{aligned} \tag{3.17}$$

The expressions for all matrices in the system have been presented previously.

Neumann conditions The Neumann conditions can be imposed in the ROM system through a penalty (see equation (3.10)) or an extra boundary term (see equation (3.11)). Since those are not used in this research, the details are omitted here.

POD-Galerkin based ROM solver: ITHACA-FV Note that the POD-Galerkin ROMs are ordinary differential equations. The system can be solved by various numerical methods, e.g., *Newton-Raphson* algorithms. In this research, the open-source library, *In real Time Highly*

Advanced Computational Applications for Finite Volumes (ITHACA-FV) [118; 368], is employed to solve the intrusive ROMs. It is developed based on OpenFOAM and provides tools for constructing and solving POD-Galerkin ROMs for parametrized problems.

It is worth noting that the library has been modified for large-scale applications in this research. The modifications include: (i) the implementation of multiple lifting functions; (ii) the extension to eddy viscosity interpolation with velocity POD coefficients; (iii) technical improvement regarding parallelization.

Validation of ROMs As explained in Section 1.3.2 and Fig. 1.6, the CFD solvers are verified and validated by experimental data. In contrast, the ROMs are validated by comparing with corresponding FOM solutions performed during the *offline* stage. Thus, the accuracy of ROMs is evaluated by comparing with high-fidelity snapshots. For example, the relative L^2 error norm for velocity is given by

$$\epsilon_{\mathbf{u}}(\mu_j) = \frac{\|\mathbf{u}(\mu_j) - \mathbf{u}_{\text{ROM}}(\mu_j)\|_{L^2(\Omega)}}{\|\mathbf{u}(\mu_j)\|_{L^2(\Omega)}}, \quad j = 1, \ldots, N_\mu, \tag{3.18}$$

where $\mathbf{u}(\mu_j)$ and $\mathbf{u}_{\text{ROM}}(\mu_j)$ are the FOM and ROM solutions, respectively. The same formulation can also be applied for pressure and eddy viscosity.

3.2.3 POD with interpolation

As discussed in Section 2.3.4.1, the essential step of PODI is to train a surrogate $\mathcal{Z}$ to map inputs (i.e., parameters) and POD coefficients. Various techniques can accomplish the task. Among them, three widely used methods are adopted in this research and are thus explained. Those are: *Radial Basis Function (RBF)*, *Gaussian Process Regression (GPR)*, and *Artificial Neural Networks (ANNs)*. They are implemented by Python libraries, e.g., *NumPy* [369], *SciPy* [370], *scikit-learn* [371], and *PyTorch* [372], respectively.

3.2.3.1 Radial Basis Function interpolation

RBF interpolation is a feasible method for constructing interpolants of high-dimensional unstructured data [373; 374]. It consists of two essential parts. First, the relations between parameters are expressed as a series of radial basis functions. The outputs are then approximated as a weighted sum of RBFs. The weighting factors are obtained in the offline stage. Details are explained as follows.

Assume that N pairs of input variables x_i and the target values $y(x_i)$ are known. Our objective is to approximate a surrogate model $\mathcal{Z}$, achieving $y(x_i) \approx \mathcal{Z}(x_i)$. Such a RBF interpolation for $y(x_i)$ is given by

$$y(x_i) = \sum_{j=1}^{N} w_j \varphi(\|x_i - x_j\|) \quad \text{with} \quad j = 1, \ldots, N, \tag{3.19}$$

where w_j are computed by solving the algebraic system. φ is the RBF kernel that is weighted by coefficients w_j, $j = 1, \ldots, N$. $\|\cdot\|$ denotes a norm. The expressions of several widely used kernels are listed in Appendix B.3.1.

Moreover, polynomial functions can be extended to enhance the capability of RBF to represent polynomial variance [373; 374; 370]. Assume an n-order polynomial function $P^n(x) = c_1 + c_2 x + \cdots + c_n x^{n-1}$, and its element $P^n_m(x) = c_m x^{m-1}$. The approximation is modified as

$$y(x_i) = \sum_{j=1}^{N} w_j \varphi(\|x_i - x_j\|) + P^n(x_i) \quad \text{with} \quad j = 1, \ldots, N,$$

in conjunction with the constraints

$$\sum_{j=1}^{N_x} w_j P^n_m(x_j) = 0 \quad \forall m = 1, \cdots, n,$$

where the unknowns are w_j and c_m. They are obtained in the offline stage and then used for prediction.

For the PODI-ROM, the RBF interpolation is applied to each POD coefficient. Coefficients w (and c if needed) are computed with the training datasets. Then, given the test parameters $\tilde{\mu}_j \in \mathcal{P},\ j = 1, \ldots, N_{\tilde{\mu}}$, the i-th POD coefficient $\alpha_i(\tilde{\mu}_j)$ is approximated by

$$\alpha_i(\tilde{\mu}_j) \approx \mathcal{Z}(\tilde{\mu}_j) = \sum_{k=1}^{N_\mu} w_k \varphi(\|\tilde{\mu}_j - \mu_k\|) + P^n(\tilde{\mu}_j),$$

where μ_k are the training parameters, and N_μ is the total number of snapshots.

3.2.3.2 Gaussian Process Regression

GPR is a probabilistic, non-parametric method used for regression and classification. A thorough description is presented in [375], and an introduction is addressed in [376].

GPR approximates observed data with a *Multivariate Normal Distribution* (MVN) [375]. Given the dataset x and its outputs $y(x)$, the distribution of $y(x)$ is written by

$$y(x) \sim \mathcal{N}\left(m(x), \Sigma\right),$$

where $m(x)$ is the mean value of x, Σ denotes the variance of x, and $\mathcal{N}$ is the operator refereed to the normal distribution.

Now, for an unknown vector x_*, GPR aims to find its prediction $y_*(x_*)$. This is achieved with the following steps.

Firstly, GPR assume that the joint distribution of $y(x)$ and $y_*(x_*)$ follows the MVN, and it is expressed as

$$\begin{bmatrix} y \\ y_* \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} m(x) \\ m(x_*) \end{bmatrix}, \begin{bmatrix} \Sigma & \Sigma_* \\ \Sigma_*^{\mathrm{T}} & \Sigma_{**} \end{bmatrix} \right),$$

where $\Sigma = \Sigma(x, x)$, $\Sigma_* = \Sigma(x, x_*)$ and $\Sigma_{**} = \Sigma(x_*, x_*)$.

Secondly, the prediction y_* follows a conditional distribution $y_* | \, y, x, x_*$. This can be evaluated by using *marginal and conditional distributions* of MVN theorem [377; 376], and the results is

$$y_* | \, y, x, x_* \sim \mathcal{N}\left(\Sigma_*^{\mathrm{T}}\Sigma^{-1}y, \Sigma_*^{\mathrm{T}}\Sigma^{-1}\Sigma_*\right).$$

To enhance generality for different scenarios, Gaussian noise with variance σ_n^2 is added to model noisy data. Such that, Σ is replaced by $\Sigma + \sigma_n^2 I$, and leads to

$$y_* | \, y, x, x_* \sim \mathcal{N}\left(\Sigma_*^{\mathrm{T}}\left[\Sigma + \sigma_n^2 I\right]^{-1} y, \Sigma_*^{\mathrm{T}}\left[\Sigma + \sigma_n^2 I\right]^{-1}\Sigma_*\right). \tag{3.20}$$

Finally, for a new input, x_*, the corresponding prediction is

$$y_* = \Sigma_*^{\mathrm{T}}[\Sigma + \sigma_n^2 I]^{-1} y.$$

Note that the variance matrix of y_* distribution is $\Sigma_*^{\mathrm{T}}[\Sigma + \sigma_n^2 I]^{-1}\Sigma_*$ [375]. The expression of variance indicates that the uncertainty bound of y_* depends only on x and x_*.

At this stage, the only missing information shown above is the formulations for computing Σ, which is called *kernel* in the frame of GPR [375; 376] (listed in Appendix B.3.2). The optimal kernel choice depends on the problem, and GPR's performance is also sensitive to this selection. Thus, numerical experiments are necessary to optimize the GPR-based surrogate.

When integrated into the PODI framework, the inputs and outputs of GPR in the offline stage are $\{\mu_j\}_{j=1}^{N_\mu}$ and $\{\boldsymbol{\alpha}(\mu_j)\}_{j=1}^{N_\mu}$, respectively. Then, for the online phase, the $\mathbf{x}_*$ given above is

replaced by the test set $\{\tilde{\mu}_j\}_{j=1}^{N_{\tilde{\mu}}}$. The prediction y_* denotes the POD coefficient $\alpha_i(\tilde{\mu}_j)$. Thus, the GPR surrogate $\mathcal{Z}$ is used as

$$\alpha_i(\tilde{\mu}_j) \approx \mathcal{Z}(\tilde{\mu}_j) = \Sigma_*^{\mathrm{T}}[\Sigma + \sigma_n^2 I]^{-1}\alpha_i, \quad i = 1, \ldots, N_\mu,$$

where N_μ and $N_{\tilde{\mu}}$ are the number of training and testing parameters, respectively.

3.2.3.3 Artificial Neural Network

The role of ANN is similar to the above two techniques in constructing PODI. [378] includes more instructions about the theory and implementation of ANN.

Recall that each neuron includes inputs x and output y as

$$y = \sigma(wx + b),$$

where σ is the activation function to entitle non-linear properties of neural networks [379] (see some common definitions in Appendix B.3.3). w and b are known as the *weight* and *bias* of the neuron, respectively.

Indeed, computing w and b is the key to training an ANN. These values are determined during the training phase by minimizing the difference between the known outputs and the predictions, which is known as the *loss function*. The mean squared error (MSE) is a widely used quantification for regression problems [354], which is defined as

$$\mathrm{MSE} = \frac{1}{N_\mu}\sum_{i=1}^{N_\mu}(y_i - \hat{y}_i)^2,$$

where y_i and $\hat{y}_i$ are the true and predicted values, respectively, and N_μ is the number of data points.

In this research, a *gradient descent-based* algorithm is employed to solve the optimization problem, i.e., *Adam* [380].

As common knowledge for the frame of neural networks, its performance highly depends on its *architecture* (e.g., the type, number, and size of the hidden layers) [354; 381]. The *activation function* should be appropriately validated to ensure the accuracy of a network [379]. The *loss function* (i.e., the objective to be minimized) is another key feature that should be considered [354]. Numerical experiments are also crucial for validating and optimizing an ANN.

The implementation of an ANN in the PODI framework is carried out similarly. The inputs $\{\mu_j\}_{j=1}^{N_\mu}$ and outputs $\{\boldsymbol{\alpha}(\mu_j)\}_{j=1}^{N_\mu}$ are applied to train the network. The POD coefficients at new parameters $\{\tilde{\mu}_j\}_{j=1}^{N_{\tilde{\mu}}}$ are predicted by the trained ANN.

3.3 Local reduced order model

An introduction and literature review about local ROMs is presented in Section 2.4. Many aspects have already been discussed. Thus, only the essential parts are explained here for completeness.

Domain decomposition is the first aspect to consider. In this research, only the non-overlapping treatment is considered. Due to the existence of repeating geometrical patterns in rod bundles, a reference block is adopted to approximate the whole domain.

Various coupling algorithms have been reported in the literature. Among all alternatives, the iterative Schwarz method can be incorporated with both intrusive and non-intrusive techniques. Given its feasibility and convenience, the *non-overlapped Dirichlet-Neumann* method is adopted in this research for constructing the local ROM.

For each sub-problem, the PODI frameworks (see Section 3.2.3) are applied. The choice is based on two reasons. Firstly, according to the comparison of both approaches (see Chapter 4), the accuracy of PODI is comparable to that of POD-Galerkin. Especially when the dataset for the offline stage is sufficient, PODI can achieve higher accuracy. Secondly, the implementation of non-intrusive approaches avoid the complexities of modifying the CFD solver.

In the following sections, a problem of two subdomains will be presented first as an example for clarification. The algorithm is then extended to multiple partitions.

3.3.1 Dirichlet-Neumann iteration for two subdomains

3.3.1.1 Two-subdomain problem setup

An example shown in Fig. 3.2 is adopted to explain the basic ideology of the Dirichlet-Neumann iteration between two non-overlapping subdomains. The global domain is $\Omega = \Omega_1 \cup \Omega_2$. The common interfaces are $\Gamma_{[12]} = \Gamma_{[21]}$. $\Gamma_{[12]} = \overline{\Omega}_1 \cap \partial\Omega_2$, and $\Gamma_{[21]} = \overline{\Omega}_2 \cap \partial\Omega_1$. Dirichlet inlet $\mathbf{u}_D$ and Neumann outlet $\mathbf{g}_N$ are enforced on Γ_D and Γ_N respectively.

To explain the interface constraints, the notations are illustrated first. The two subscripts, $_{[12]}$ and $_{[21]}$, are applied to indicate the direction of information exchange. For example, in $\Gamma_{[12]}$, variables $\mathbf{u}_{[12]}$ and their surface normal gradients (i.e., flux) $\mathbf{g}_{[12]} = \nabla\mathbf{u}_{[12]} \cdot \mathbf{n}_{[12]}$ are given, where $\mathbf{n}$ is the unit normal vector pointing from Ω_1 to Ω_2. Similarly, $\mathbf{u}_{[21]}$ and $\mathbf{g}_{[21]} = \nabla\mathbf{u}_{[21]} \cdot \mathbf{n}_{[21]}$ are defined on $\Gamma_{[21]}$, where $\mathbf{n}_{[21]} = -\mathbf{n}_{[12]}$. The continuity at $\Gamma_{[12]} = \Gamma_{[21]}$ ensures $\mathbf{u}_{[12]} = \mathbf{u}_{[21]}$ and $\mathbf{g}_{[12]} = -\mathbf{g}_{[21]}$.

Dirichlet and Neumann conditions are applied for the two sub-problems. For Ω_1, they are $\mathbf{u}_D$ and $\mathbf{g}_{[12]}$, and those for Ω_2 are $\mathbf{u}_{[21]}$ and $\mathbf{g}_N$. Note that, for non-overlapping partitions, Dirichlet-Neumann conditions on the common faces are required to ensure that the assembly of the local solutions equals the global results. See explanations in [285].

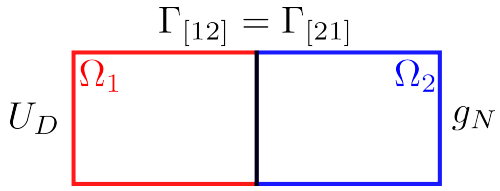


Figure 3.2: A domain consists of two non-overlapping subdomains Ω_1 and Ω_2 and their common interface $\Gamma_{[12]} = \Gamma_{[21]}$. The Dirichlet $\mathbf{u}_D$ and Neumann $\mathbf{g}_N$ conditions are assigned at the inlet and outlet faces, respectively.

For the aforementioned problem, the procedures for computing subdomain-level POD modes and constructing local PODI ROMs are explained in the following subsections.

3.3.1.2 Decomposition and POD modes

In the Dirichlet-Neumann iteration, two types of POD modes are required: cell- and boundary-based modes. They are used to approximate solutions in the subdomain and at the interface, respectively.

The cell-based basis functions are obtained as follows. The FVM solutions are collected as snapshot matrices $\mathbf{S}_1 = \left[\mathbf{u}_{1,1}, \ldots, \mathbf{u}_{1,N_\mu}\right]$ and $\mathbf{S}_2 = \left[\mathbf{u}_{2,1}, \ldots, \mathbf{u}_{2,N_\mu}\right]$. POD is applied for their reduction and results in

$$\mathbf{S}_1 = \Phi_1 A_1, \text{ and } \mathbf{S}_2 = \Phi_2 A_2,$$

where Φ_1 and Φ_2 store modes in column-wise, and A_1 and A_2 contain the POD coefficients.

The above modes consist of values only in cell-centers, and now, let us illustrate the procedure for computing boundary values for the cell-based basis. The snapshots at $\Gamma_{[12]}$ and $\Gamma_{[21]}$ denote $\mathbf{S}_{[12]} = \left[\mathbf{u}_{[12],1}, \ldots, \mathbf{u}_{[12],N_\mu}\right]$ and $\mathbf{S}_{[21]} = \left[\mathbf{u}_{[21],1}, \ldots, \mathbf{u}_{[21],N_\mu}\right]$, respectively. The same for the flux, i.e., $\mathbf{G}_{[12]} = \left[\mathbf{g}_{[12],1}, \ldots, \mathbf{g}_{[12],N_\mu}\right]$ and $\mathbf{G}_{[21]} = \left[\mathbf{g}_{[21],1}, \ldots, \mathbf{g}_{[21],N_\mu}\right]$. Assuming the interface quantities are reduced considering the same coefficients, and one can obtain

$$\begin{aligned} \mathbf{S}_{[12]} &= \Phi_{[12]} A_1, \quad \mathbf{G}_{[12]} = \Psi_{[12]} A_1. \\ \mathbf{S}_{[21]} &= \Phi_{[21]} A_2, \quad \mathbf{G}_{[21]} = \Psi_{[21]} A_2. \end{aligned}$$

Be aware that, due to the continuity across interfaces, the equalities hold, i.e., $\mathbf{S}_{[12]} = \mathbf{S}_{[21]}$ and $\mathbf{G}_{[12]} = -\mathbf{G}_{[21]}$.

The next step is to compute the face-based POD modes. The datasets are collected as $\mathbf{G}_{[12]}$ and $\mathbf{S}_{[21]}$. POD is applied to and yields

$$\mathbf{G}_{[12]} = \Psi^*_{[12]} A^*_{[12]}, \text{ and } \mathbf{S}_{[21]} = \Phi^*_{[21]} A^*_{[21]}.$$

Note that the values are given considering the boundary condition of the two sub-problems. $\mathbf{g}_{[12]}$ is the Neumann condition for Ω_1 and $\mathbf{u}_{[21]}$ is Dirichlet inlet for Ω_2.

3.3.1.3 Local boundary value solver

Once the basis functions are obtained, the local PODI model can be constructed for each subdomain. The details for PODI can be seen from Section 3.2.3. Considering the Dirichlet and/or Neumann conditions at the interface, each local reduced system can be expressed as a *Boundary Value Solver*. Surrogate models are trained to approximate the mapping from boundary conditions (and other parameters) to outputs (i.e., POD coefficients).

The construction steps are summarized as:

1. Assume the parameters for Ω_1 and Ω_2 are denoted as $\boldsymbol{\mu}_1 = \left[\mu_{1,1}, \ldots, \mu_{1,N_\mu}\right]$ and $\boldsymbol{\mu}_2 = \left[\mu_{2,1}, \ldots, \mu_{2,N_\mu}\right]$. If the global boundary conditions are parametrized, they are defined as $\mathbf{u}_D = \mathbf{u}_D(\boldsymbol{\mu}_D)$ and $\mathbf{g}_N = \mathbf{g}_N(\boldsymbol{\mu}_N)$.
2. In the offline stage, one can obtain the training datasets $\mathbf{S}_1$ and $\mathbf{S}_2$. Applying the definitions of modes shown above, the cell-based POD coefficients are $\boldsymbol{\alpha}_1$ and $\boldsymbol{\alpha}_2$ for Ω_1 and Ω_2, respectively. Note that the boundary conditions for Ω_1 and Ω_2 are $\mathbf{u}_D$ and $\mathbf{g}_{[12]}$, and $\mathbf{u}_{[21]}$ and $\mathbf{g}_N$, respectively. Thus, the interface snapshots are $\mathbf{G}_{[12]}$ and $\mathbf{S}_{[21]}$. Their modes and coefficients are $\Phi^*_{[21]}$ and $\Psi^*_{[12]}$, and $\boldsymbol{\alpha}^*_{[21]}$ and $\boldsymbol{\alpha}^*_{[12]}$.
3. Two PODI models for the two partitions are trained as

$$\mathcal{Z}_1 : \left(\boldsymbol{\mu}_D, \boldsymbol{\alpha}^*_{[12]}, \boldsymbol{\mu}_1\right) \mapsto \boldsymbol{\alpha}_1, \text{ and } \mathcal{Z}_2 : \left(\boldsymbol{\alpha}^*_{[21]}, \boldsymbol{\mu}_N, \boldsymbol{\mu}_2\right) \mapsto \boldsymbol{\alpha}_2. \tag{3.21}$$

   $\boldsymbol{\alpha}_1$ and $\boldsymbol{\alpha}_2$ can be reconstructed to the full solutions by $\mathbf{u}_1 = \Phi_1\boldsymbol{\alpha}_1$ and $\mathbf{u}_2 = \Phi_2\boldsymbol{\alpha}_2$.

3.3.1.4 Interface mapping

Another aspect should be noted is how to obtain $\boldsymbol{\alpha}^*_{[12]}$ and $\boldsymbol{\alpha}^*_{[21]}$ when $\boldsymbol{\alpha}_2$ and $\boldsymbol{\alpha}_1$ are available. They can be obtained via two approaches, i.e., *projection-based* and *Regression-based* methods.

The projection-based approach consists of several steps: (i) Given $\boldsymbol{\alpha}_2$, flux at $\Gamma_{[21]}$ denotes $\mathbf{g}_{[21]} = \Psi_{[21]}\boldsymbol{\alpha}_2$; (ii) set $\mathbf{g}_{[12]} = -\mathbf{g}_{[21]}$ (interface continuity); (iii) Project $\mathbf{g}_{[12]}$ onto the face basis to obtain $\boldsymbol{\alpha}^*_{[12]} = (\Psi^*_{[12]})^{\mathrm{T}}\mathbf{g}_{[12]}$. Consequently, combining the above steps, one has $\boldsymbol{\alpha}^*_{[12]} = -(\Psi^*_{[12]})^{\mathrm{T}}\Psi_{[21]}\boldsymbol{\alpha}_2$. Similarly, $\boldsymbol{\alpha}^*_{[12]}$ is computed by $\boldsymbol{\alpha}^*_{[21]} = (\Phi^*_{[21]})^{\mathrm{T}}\Phi_{[12]}\boldsymbol{\alpha}_1$.

In contrast, the regression-based strategy is more straightforward. Two extra models are trained during the offline stage as

$$\mathcal{Z}_{[12]} : \boldsymbol{\alpha}_2 \mapsto \boldsymbol{\alpha}^*_{[12]}, \text{ and } \mathcal{Z}_{[21]} : \boldsymbol{\alpha}_1 \mapsto \boldsymbol{\alpha}^*_{[21]}.$$

Note that the regression-based method can handle non-conforming interface grids, whereas the former requires matching meshes at the interface.

3.3.1.5 The coupling of sub-problems

When the reduced bases and two boundary value solvers are available, the next step is to couple them via the Dirichlet-Neumann iteration, as shown in Algorithm 1.

Algorithm 1 Non-intrusive Local Reduced Order Model via Iterative Dirichlet-Neumann Algorithm for two subdomains

Require:
- **Boundary conditions:** Dirichlet $\mathbf{u}_D$ and Neumann $\mathbf{g}_N$
- **Parameters:** $\boldsymbol{\mu}_1, \boldsymbol{\mu}_2$ for subdomains 1 and 2
- **Cell-based basis:** Φ_1, Φ_2 and $\Phi_{[12]}$ (value), $\Psi_{[21]}$ (flux)
- **Face-based basis:** $\Psi^*_{[12]}$, $\Phi^*_{[21]}$
- **Trained surrogate models:** $\mathcal{Z}_1 : (\mathbf{u}_D, \boldsymbol{\alpha}^*_{[12]}, \boldsymbol{\mu}_1) \mapsto \boldsymbol{\alpha}_1$; $\mathcal{Z}_2 : (\boldsymbol{\alpha}^*_{[21]}, \mathbf{g}_N, \boldsymbol{\mu}_2) \mapsto \boldsymbol{\alpha}_2$
- **Algorithm parameters:** $\lambda \in (0, 1]$ (relaxation factor), $\varepsilon_0 > 0$ (tolerance), $k_{\max}$ (max iterations), $\theta \in \{0, 1\}$ ($\theta = 0$: additive Schwarz, $\theta = 1$: multiplicative Schwarz)
- **Initial guesses:** $\boldsymbol{\alpha}^{*,0}_{[12]}$; $\boldsymbol{\alpha}^{*,0}_{[21]}$; $\boldsymbol{\alpha}^0_1$; $\boldsymbol{\alpha}^0_2$

Initialize: $k \leftarrow 0$, converged $\leftarrow$ false
repeat
  Subdomain 1:
  $\boldsymbol{\alpha}^{*,k+1}_{[12]} \leftarrow -(\Psi^*_{[12]})^{\mathrm{T}} \Psi_{[21]} \boldsymbol{\alpha}^k_2$ ▷ Obtain interface condition
  $\boldsymbol{\alpha}^{*,k+1}_{[12]} = \lambda \boldsymbol{\alpha}^{*,k+1}_{[12]} + (1 - \lambda) \boldsymbol{\alpha}^{*,k}_{[12]}$ ▷ Relaxed update
  $\boldsymbol{\alpha}^{k+1}_1 \leftarrow \mathcal{Z}_1(\mathbf{u}_D, \boldsymbol{\alpha}^{*,k+\theta}_{[12]}, \boldsymbol{\mu}_1)$ ▷ $\theta = 0$ (add.) or $\theta = 1$ (multi.) Schwarz
  Subdomain 2:
  $\boldsymbol{\alpha}^{*,k+1}_{[21]} \leftarrow (\Phi^*_{[21]})^{\mathrm{T}} \Phi_{[12]} \boldsymbol{\alpha}^{k+1}_1$ ▷ Obtain interface condition
  $\boldsymbol{\alpha}^{*,k+1}_{[21]} = \lambda \boldsymbol{\alpha}^{*,k+1}_{[21]} + (1 - \lambda) \boldsymbol{\alpha}^{*,k}_{[21]}$ ▷ Relaxed update
  $\boldsymbol{\alpha}^{k+1}_2 \leftarrow \mathcal{Z}_2(\boldsymbol{\alpha}^{*,k+\theta}_{[21]}, \mathbf{g}_N, \boldsymbol{\mu}_2)$ ▷ $\theta = 0$ (add.) or $\theta = 1$ (multi.) Schwarz
  Convergence check:
  $\varepsilon^k_g \leftarrow \dfrac{\|\boldsymbol{\alpha}^{*,k+1}_{[12]} - \boldsymbol{\alpha}^{*,k}_{[12]}\|_{L^2(\Gamma_{[12]})}}{\|\boldsymbol{\alpha}^{*,k}_{[12]}\|_{L^2(\Gamma_{[12]})}}$ ▷ Relative value error
  $\varepsilon^k_u \leftarrow \dfrac{\|\boldsymbol{\alpha}^{*,k+1}_{[21]} - \boldsymbol{\alpha}^{*,k}_{[21]}\|_{L^2(\Gamma_{[21]})}}{\|\boldsymbol{\alpha}^{*,k}_{[21]}\|_{L^2(\Gamma_{[21]})}}$ ▷ Relative flux error
  $\varepsilon^k \leftarrow \max(\varepsilon^k_u, \varepsilon^k_g)$
  $k \leftarrow k + 1$
  if $\varepsilon^k < \varepsilon_0$ **then**
    converged $\leftarrow$ true
  end if
until converged $=$ true **or** $k \geq k_{\max}$
return $\mathbf{u}_1 = \Phi_1 \boldsymbol{\alpha}^k_1$, $\mathbf{u}_2 = \Phi_2 \boldsymbol{\alpha}^k_2$, final iteration k, convergence status

Now, let us explain the implementation of this algorithm. The main idea is to solve two subproblems iteratively until convergence.

- In the Dirichlet step, inputs are inlet $\mathbf{u}_D$, the outlet $\boldsymbol{\alpha}^*_{[12]}$ and $\boldsymbol{\mu}_1$. Through $\mathcal{Z}_1$, $\boldsymbol{\alpha}_1$ is obtained. Then, interface mapping is applied to get $\boldsymbol{\alpha}^*_{[21]}$.
- In the Neumann step, $\mathbf{u}_2$ is computed via $\mathcal{Z}_2$ with inputs $\boldsymbol{\alpha}^*_{[21]}$, $\mathbf{g}_N$, and $\boldsymbol{\mu}_2$. Then, $\boldsymbol{\alpha}^*_{[12]}$ is updated through the mapping.

As indicated in Algorithm 1, a relaxation factor λ is applied to enhance stability and convergence of the algorithm [285]. Like the Schwarz method, both multiplicative ($\theta = 1$) and additive ($\theta = 0$) procedures are feasible for the iteration. Note that only the projection-based approach for interface mapping is illustrated. The regression-based method can be easily integrated by replacing the projection steps.

3.3.2 Dirichlet-Neumann iteration for multiple-subdomain problems

Now, let us extend the above procedure to multiple subdomains. Different strategies can be found in the literature for domain decomposition (see Section 2.4). Hereafter, two scenarios are explained, i.e., individual and generic decomposition.

3.3.2.1 Individual decomposition

When the individual decomposition is considered, the procedure is straightforward. Suppose the global domain Ω is decomposed into N_Ω non-overlapping subdomains, i.e., $\Omega = \bigcup_{m=1}^{N_\Omega} \Omega_m$. The interface between Ω_m and Ω_n is $\Gamma_{[mn]} = \overline{\Omega}_m \cap \partial\Omega_n$ and $\Gamma_{[nm]} = \overline{\Omega}_n \cap \partial\Omega_m$. Since only non-overlapping partitions are considered, $\Gamma_{[mn]} = \Gamma_{[nm]}$. Note that $\overline{\Omega}_m$ denotes the closure of domain, i.e., $\overline{\Omega}_m = \Omega_m \cup \partial\Omega_m$.

A set of modes is computed for the divisions and shared faces, i.e., Φ_m for Ω_m, and $\Phi_{[mn]}$, $\Psi_{[mn]}$ for $\Gamma_{[mn]}$. The face-based modes are $\Phi^*_{[mn]}$ and $\Psi^*_{[mn]}$. Additionally, surrogates $\mathcal{Z}_m, m = 1, \ldots, N_\Omega$ are trained. Then, by extending Algorithm 1 to N_Ω local problems, the coupling for multiple subdomains is achieved, as summarized in Algorithm 2. Note that the steps different from Algorithm 1 are highlighted in blue.

3.3.2.2 Generic decomposition

Another situation is when several generic subdomains assemble the global domain. Algorithm 2 still applies to the case, except that the basis functions and local solvers are modified.

The generic partitions denote $\hat{\Omega}^k$, $k = 1, \ldots, \hat{N}_\Omega$, and the total amount of subdomains is N_Ω. Fig. 3.3b presents a sketch of a reference block $\hat{\Omega}^1$ and its faces $\hat{\Gamma}^1$ and $\hat{\Gamma}^2$.

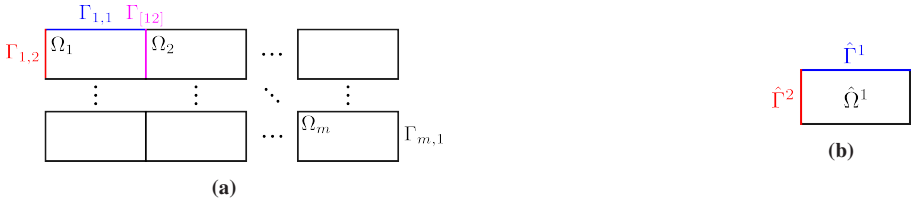


Figure 3.3: Decomposition of a domain and the reference subdomain. (a) A global domain consists of multiple non-overlapping subdomains $\Omega_1, \Omega_2, \ldots, \Omega_{N_\Omega}$ and interfaces $\Gamma_{[mn]} = \overline{\Omega}_m \cap \partial\Omega_n$. (b) A reference domain $\hat{\Omega}^1$, and faces $\hat{\Gamma}^1$ and $\hat{\Gamma}^2$.

Since the whole domain is assembled by instantiations of the generic partitions, the way to compute basis functions and local boundary value solvers should be modified. They are explained as follows.

Firstly, the solutions collected from all instantiations of $\hat{\Omega}^k$ are stacked to compute the cell-based basis $\hat{\Phi}^k$. For example, Ω_1^k, Ω_2^k and Ω_3^k are transformed from $\hat{\Omega}^k$. N_μ computations are performed. The snapshots matrix is $\mathbf{S}^k = \left[\mathbf{u}_{1,1}^k, \ldots, \mathbf{u}_{1,N_\mu}^k, \mathbf{u}_{2,1}^k, \ldots, \mathbf{u}_{2,N_\mu}^k, \mathbf{u}_{3,1}^k, \ldots, \mathbf{u}_{3,N_\mu}^k\right]$. POD is applied to extract modes from $\mathbf{S}^k$. Thus, the number of snapshots is $\hat{N}_\Omega$, and $\hat{N}_\Omega$ sets of cell-based bases $\hat{\Phi}^k$ are obtained. The treatment is similar for boundaries. For instance, for a face $\Gamma_j^k \in \partial\hat{\Omega}^k$, $\hat{\Phi}_j^k$, and $\hat{\Psi}_j^k$ are used to approximate the solutions and fluxes at the interfaces. The superscript k and subscript j indicate that Γ_j^k is the j-th outer face of $\hat{\Omega}^k$.

Secondly, face-based snapshots are collected, and RBs are computed. Suppose $\hat{N}_\Gamma$ types of interfaces exist, and they are defined as $\hat{\Gamma}^r$, $r = 1, \ldots, \hat{N}_\Gamma$. Fields at all faces transformed from $\hat{\Gamma}^r$ are gathered to compute the boundary-based bases $\hat{\Phi}^{r,*}$ and $\hat{\Psi}^{r,*}$. Note that $\Gamma_j^k \in \partial\hat{\Omega}^k$ is an instantiation of a reference face. To better clarify the relationship between the subdomain and the face, Γ_j^k is used instead of Γ_i^r (i-th instantiation).

As for the local boundary value solvers, only $\hat{N}_\Omega$ surrogates are trained, indeed, $\hat{\mathcal{Z}}^k, k = 1, \ldots, \hat{N}_\Omega$. When solving an instantiation Ω_m transformed from $\hat{\Omega}^k$, the $\hat{\mathcal{Z}}^k$ is applied. For boundaries $\Gamma_j^k \in \partial\hat{\Omega}^k$, $\hat{\Phi}_j^k$, and $\hat{\Psi}_j^k$ are used to approximate the solutions and fluxes at the interfaces.

In short, Algorithm 2 still holds, while the RBs $\hat{\Phi}^k$, $\hat{\Phi}^{r,*}$, $\hat{\Psi}^{r,*}$, and the surrogates $\hat{\mathcal{Z}}^k$ are employed.

Algorithm 2 Non-intrusive Local Reduced Order Model for Multiple Subdomains via Iterative Dirichlet-Neumann Algorithm.

Require:
- **Domain decomposition:** N_Ω subdomains with interfaces $\Gamma_{[mn]} = \partial\Omega_m \cap \partial\Omega_n$. $\mathcal{N}_\Gamma(m)$ denotes the set of neighbors of subdomain m.
- **Boundary conditions:** Global Dirichlet $\mathbf{u}_D$ and Neumann $\mathbf{g}_N$
- **Parameters:** $\boldsymbol{\mu}_m$ for each subdomain m
- **Cell-based basis:** Φ_m for $m = 1, \ldots, N_\Omega$ (and $\Phi_{[mn]}$, $\Psi_{[mn]}$ for interface $\Gamma_{[mn]}$ if projection-based mapping is used)
- **Face-based basis:** $\Psi^*_{[mn]}$, $\Phi^*_{[mn]}$ (and $\mathcal{Z}_{[mn]}$ for regression-based mapping) for $\Gamma_{[mn]}$
- **Trained surrogate models:** $\mathcal{Z}_m : (\{\boldsymbol{\alpha}^*_{[mj]}\}_{j\in\mathcal{N}_\Gamma(m)}, \boldsymbol{\mu}_m) \mapsto \boldsymbol{\alpha}_m$ for $m = 1, \ldots, N_\Omega$
- **Algorithm parameters:** $\lambda \in (0, 1]$ (relaxation), $\varepsilon_0 > 0$ (tolerance), $k_{\max}$ (max iterations), $\theta \in \{0, 1\}$ ($\theta = 0$: additive Schwarz, $\theta = 1$: multiplicative Schwarz)
- **Initial guesses:** $\boldsymbol{\alpha}^0_m$ for all subdomains, $\boldsymbol{\alpha}^{*,0}_{[mn]}$ for all interfaces

Initialize: $k \leftarrow 0$, converged $\leftarrow$ false
repeat
  for $m = 1$ to N_Ω **do** ▷ Loop over all subdomains
    Step 1: Solve subdomain m using surrogate model
    $\boldsymbol{\alpha}^k_m \leftarrow \mathcal{Z}_m(\{\boldsymbol{\alpha}^{*,k}_{[mj]}\}_{j\in\mathcal{N}_\Gamma(m)}, \boldsymbol{\mu}_m)$
    Step 2: Interface mapping
    for $j \in \mathcal{N}_\Gamma(m)$ **do** ▷ Loop over neighbors $\mathcal{N}_\Gamma(m)$
      $\boldsymbol{\alpha}^{*,k+1}_{[jm]} \leftarrow (\Phi^*_{[jm]})^{\mathrm{T}}\Phi_{[mj]}\boldsymbol{\alpha}^k_m$ or $\mathcal{Z}_{[jm]}(\boldsymbol{\alpha}^k_j)$ or $\mathbf{u}_D$ ▷ Dirichlet conditions
      $\boldsymbol{\alpha}^{*,k+1}_{[jm]} \leftarrow -(\Psi^*_{[jm]})^{\mathrm{T}}\Psi_{[mj]}\boldsymbol{\alpha}^k_m$ or $\mathcal{Z}_{[jm]}(\boldsymbol{\alpha}^k_j)$ or $\mathbf{g}_D$ ▷ Neumann conditions
    end for
    Step 3: Update and relaxation
    for $j \in \mathcal{N}_\Gamma(m)$ **do**
      $\boldsymbol{\alpha}^{*,k+1}_{[jm]} \leftarrow \lambda\boldsymbol{\alpha}^{*,k+1}_{[jm]} + (1-\lambda)\boldsymbol{\alpha}^{*,k}_{[jm]}$
    end for
  end for
  Convergence check:
  for all interfaces $\Gamma_{[mn]}$ **do**
    $\varepsilon^k \leftarrow \dfrac{\|\boldsymbol{\alpha}^{*,k+1}_{[mn]} - \boldsymbol{\alpha}^{*,k}_{[mn]}\|_{L^2(\Gamma_{[mn]})}}{\|\boldsymbol{\alpha}^{*,k}_{[mn]}\|_{L^2(\Gamma_{[mn]})}}$ ▷ Relative error
  end for
  $k \leftarrow k + 1$
  if $\varepsilon^k < \varepsilon_0$ **then**
    converged $\leftarrow$ true
  end if
until converged = true **or** $k \geq k_{\max}$
return $\{\mathbf{u}_m = \Phi_m\boldsymbol{\alpha}^k_m\}_{m=1}^{N_\Omega}$, final iteration k, convergence status

4 Global ROM approach for flow analysis of rod bundle geometries

This chapter will describe the applications of global ROM (both *intrusive* and *non-intrusive*) for the steady state Reynolds-averaged Navier-Stokes equation. Numerical tests are carried out on three typical rod bundle structures: a single subchannel, a 7-pin bare rod bundle, and a 7-pin wire-wrapped bundle.

The analysis of each case is organized as follows. The CFD/FOM simulations (i.e., the *offline* stage) are discussed first. The geometry, physical setups, and parameters are indicated. High-fidelity solutions are presented and briefly analyzed. Then, the implementation of ROMs (i.e., *online* stage) is presented. FOM results are regarded as a reference for evaluating the performance and accuracy of various ROMs. As illustrated in Chapter 3, many configurations can be adapted when constructing ROMs. Thus, a comprehensive analysis of the effects of different factors is performed, as summarized in Table 4.1. The factors are categorized into two types: common and specific. The common factors are the size of the training data and the number of POD modes used to build the ROMs. Those two aspects are determined first. Then, the settings of different ROM techniques are described. For the POD-Galerkin approaches, two factors are considered: boundary condition treatment and stabilization approaches. For PODI techniques, the effect of different interpolation techniques are compared. Besides, within each interpolation algorithm, the influences of hyperparameters and/or setups are discussed. In total, more than 800 ROM computations are performed to analyze the effects of different factors and configurations.

Table 4.1: Comparison of POD-Galerkin and POD with interpolation methods.

Category	POD-Galerkin		POD with interpolation	
Methods	Stabilization	Supremizer	RBF / GPR	Kernel
		Pressure Poisson Eq.		Hyperparameter
	Dirichlet BCs	Lifting functions	ANN	Layers neurons
		Penalty		Activation functions
Effects	Number of POD modes			
	Size of the training set			

4.1 A single subchannel case

4.1.1 High fidelity simulations

The FOM computations are described highlighting several essential parts. The description begins with the geometrical details: geometry, boundary conditions, and parameter range. Then, mesh and numerical configurations are introduced, followed by visualization and discussion of high-fidelity solutions.

4.1.1.1 Physical setups

As indicated in Section 1.2, subchannels are the basic element of a rod bundle in the frame of lumped parameter analysis. Thus, the numerical test is first performed in this simple model. Note that the decomposition of rod bundles is already shown in Figs. 1.7 and 1.8.

The 3D plot of an inner subdomain is displayed in Fig. 4.1. The size of the simulation domain is determined by referring to [382; 383]. The rod diameter D and pitch P (distances between two neighbouring rod centroids) are $8.2\ \mathrm{mm}$, $10.49\ \mathrm{mm}$, respectively. That results in a hydraulic diameter of $D_h = 7.0\ \mathrm{mm}$. The axial length $L = 200\ \mathrm{mm}$, which ensure $L > 30 \times D_h$. The value is determined to ensure the flow is fully developed at the channel exit, according to an empirical correlation [384], $L/D_h \geq 4.4\, Re^{1/6}$.

As displayed in Fig. 4.1, the fluid flows along the longitudinal direction (i.e., z-axis) from an inlet (Dirichlet boundary, Γ_{in}) to an outlet (homogeneous Neumann condition, Γ_{out}). The rod surface is assigned as the no-slip wall ($\mathbf{u}|_{\Gamma_{\text{wall}}} = \mathbf{0}$). The faces to its neighbours are symmetric ($\frac{\partial \mathbf{u}}{\partial \mathbf{n}}\big|_{\Gamma_{\text{sym}}} = \mathbf{0}$), as suggested from the references [65; 66; 72; 73].

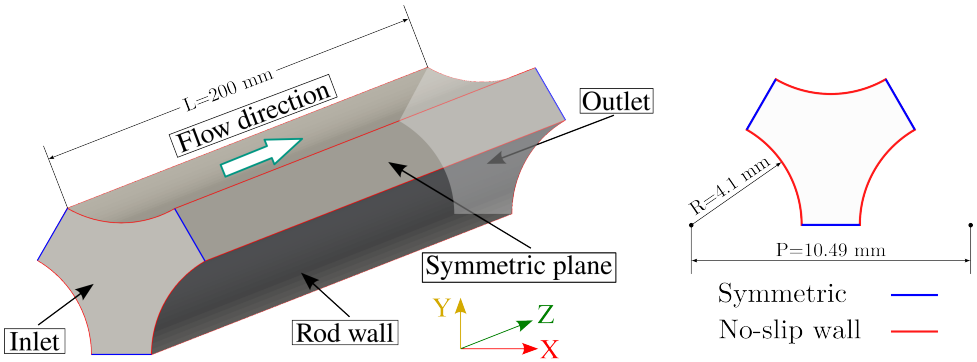


Figure 4.1: Computational domain of the inner subchannel. Left: 3D model and boundaries. Right: boundary conditions.

For the flow in a rod bundle, two parameters are relevant, i.e., inlet velocity and viscosity (e.g., in [382]). The bulk Reynolds number (Re) varies due to the two values. CFD simulations

are carried out in a specific range of Re regarding the operation condition of reactors (see [67; 56; 69; 385]).

In the simulation, heat transfer is not modeled, and kinematic viscosity ν is treated as an independent input. Additionally, the inlet velocity in the z direction is considered as a sum of two profiles: a uniform u_a profile and a power-law u_p profile. $u_a = 1\,\mathrm{m/s}$ and the area-averaged $\overline{u}_p = 1\,\mathrm{m/s}$. The total inlet velocity is then expressed as $u_{\text{in}} = \mu_1^L u_a + \mu_2^L u_p$.

The two inflows are plotted in Fig. 4.2. The non-uniform inlet velocity is characterized by the *one seventh power law*, namely $\frac{u_p}{u_{\max}} = \left(\frac{d}{d_{\max}}\right)^{1/7}$, where d denotes *distance* to the wall. $u_{\max}$ is adjusted to ensure $\overline{u}_p = 1\,\mathrm{m/s}$. Note that the $1/N$-power law is an empirical relationship used to describe the variance of velocity components normal to a wall in turbulent boundary layers. The chosen value, $1/7$, is reasonable a turbulent in-pipe flow [386].

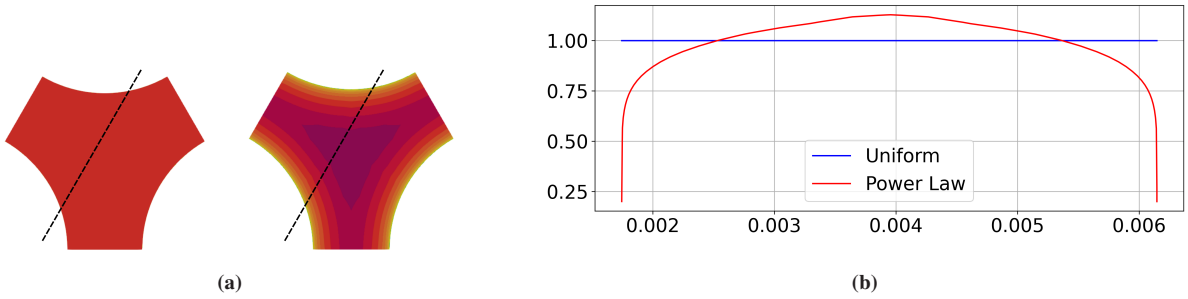


Figure 4.2: Inflow distribution for the inner subchannel. Left: Contours of uniform and the *one seventh power law* ($\frac{\mathbf{u}_p}{u_{\max}} = \left(\frac{d}{d_{\max}}\right)^{1/7}$) inflows. Right: the velocity values along a line (the black dashed line on the left).

In short, three parameters (i.e., ν, μ_1^L and μ_2^L) are involved in the following ROM simulations. Their ranges, listed in Table 4.2, are determined referring to [382; 383], which present the operating conditions of a liquid metal-cooled test facility.

Table 4.2: The range of flow parameters, referring to [382; 383].

Parameter	Unit	Min.	Max.
Mean velocity, $\overline{u}_{\text{in}}$	$\mathrm{m/s}$	0.6	2.1
Kinematic viscosity ν	$\mathrm{m^2/s}$	1.486×10^{-7}	2.327×10^{-7}
Reynolds number, Re	-	1.4×10^4	6.0×10^4

In total, 50 points (the number is noted as N_μ) are generated randomly regarding a uniform distribution within the ranges, as plotted in Fig. D.1. They can be considered as vectors in the parameter spaces. Be aware that μ_1^L, μ_2^L, and ν are constrained to ensure that Re remains within the desired limits (see Table 4.2).

The 50 corresponding FOM solutions are divided into two datasets, namely the train and test sets, to build and validate ROMs [145]. Their sizes are N_{train} and N_{test}. The training samples

are selected from the whole dataset, exploiting the *greedy* algorithm, as shown in Algorithm 3. Note that different N_{train} (e.g., 5, 10, 20, 30) are employed, but N_{test} is fixed to 20.

4.1.1.2 Simulations and results

A structured mesh is generated for the model, as shown in Fig. 4.3. The total element size is $396,000$. The thickness of boundary cells is assigned to ensure $y^+ \approx 1$, which is required when employing the SST $k-\omega$ model to resolve the velocity distribution at the boundary layer. The numerical schemes and solvers in the frame of FVM for the simulations are listed in Tables C.1 and C.2.

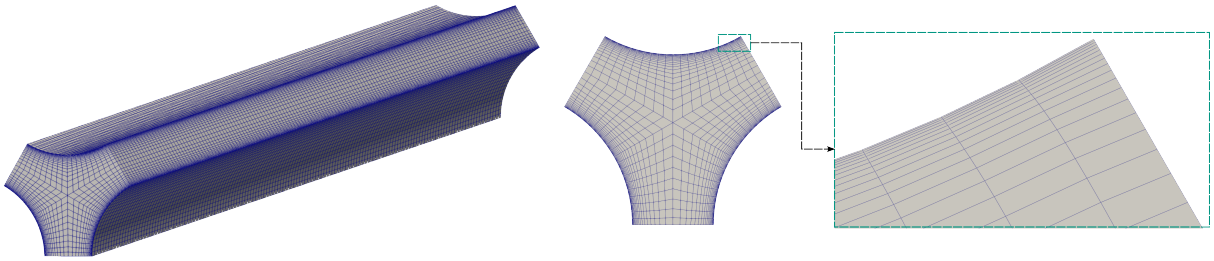


Figure 4.3: Structured mesh of the subchannel. Left: global grid; Middle: mesh of a cross-section; Right: boundary cells.

In a CFD simulation, many outputs can be analyzed. Here, only the velocity fields are presented and discussed for brevity. Three contours of velocity under different inflows are plotted in Fig. 4.4. Their inlets are power-law, nearly uniform, or combined inflows. The entrance region is highly affected by the inlet velocity. Then, the flow develops along the longitudinal direction. In all scenarios, the flow is almost fully developed at the exit boundary, with similar profiles (the values are not identical). Be aware that the flow is formulated by RANS, and thus the resulting fields are assumed to be symmetric for symmetric geometrical and physical conditions. It is easy to recognize that dominant flow patterns exist. That can be seen from POD modes.

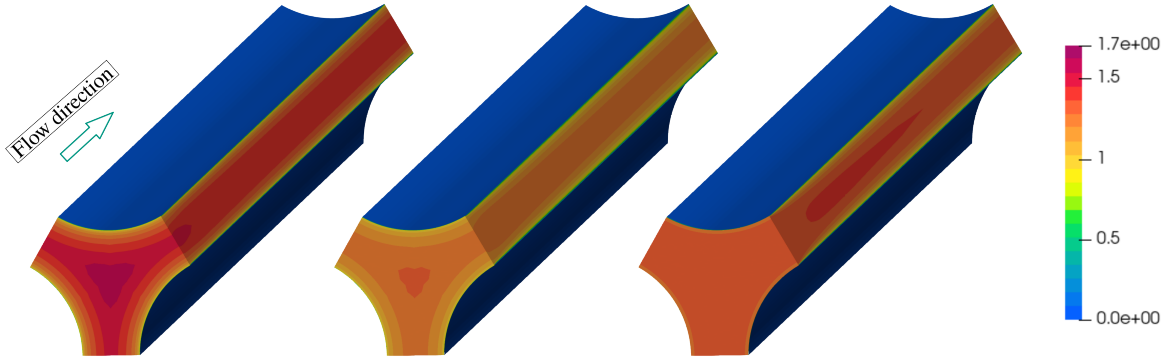


Figure 4.4: Velocity contours of three cases. Left: mostly power-law, $\mu_1^L = 0.098$, $\mu_2^L = 1.276$, and $\nu = 1.72 \times 10^{-7}$ m^2/s. Middle: combined, $\mu_1^L = 0.498$, $\mu_2^L = 0.683$, and $\nu = 2.11 \times 10^{-7}$ m^2/s. Right: nearly uniform, $\mu_1^L = 1.214$, $\mu_2^L = 0.075$, and $\nu = 1.88 \times 10^{-7}$ m^2/s.

Since this research does not focus on the high-fidelity simulations, the results are not further detailed here. The following sections will present the construction and implementation of ROMs.

4.1.2 Reduced order models

After the offline stage, both POD-Galerkin and POD with interpolation are applied to build ROMs. The following subsections present the POD modes and the results of the intrusive and non-intrusive ROMs.

4.1.2.1 POD and dominant features

Now, let us start with the POD analysis. Firstly, 30 training samples (i.e., $N_{\text{train}} = 30$) are selected using Algorithm 3 from the 50 points in the parameter set $\mathcal{P}$, as displayed in Fig D.6. Then, dominant modes are extracted from the training set.

Note that the velocity fields are homogenized with lifting functions before performing POD. Two solutions with $\mu_1^L = 1, \mu_2^L = 0$ and $\mu_1^L = 0, \mu_2^L = 1$ are used to lift the snapshots, as shown in Fig. 4.5. They represent the uniform and power-law inflow features, respectively. It can be seen that the development of the bulk flow is somehow involved in both fields. Consequently, the POD modes are generally applied to approximate the effects of viscosity and inflow profiles.

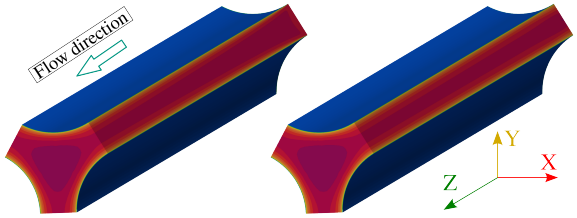


Figure 4.5: Two lifting functions with Dirichlet condition characterized with $\mu_1^L = 1, \mu_2^L = 0$ (left) and $\mu_1^L = 0, \mu_2^L = 1$ (right).

The variance of the POD truncated energy norm ($\mathcal{E}^2$, see equation (3.7)) against the number of modes is plotted in Fig. 4.6, consisting of three lines for velocity, pressure, and eddy viscosity, respectively. As shown in the figure, the decay of $\mathcal{E}^2$ is nearly exponential, indicating the existence of dominant features. The line reveals that the first pressure mode captures more than 99.99% of the total energy. The decay of the remaining two fields is similar but slower than that of the pressure. They can reach 99.99% before the 10^{th} mode. Thus, the size of the POD basis can be determined depending on the desired accuracy.

The contours of the first four POD modes of velocity are displayed in Fig. 4.7. The bulk flow is already eliminated with the two lifting functions. The dominant features can be interpreted as follows. The first mode aims to account for the near-wall behaviors. The second and third

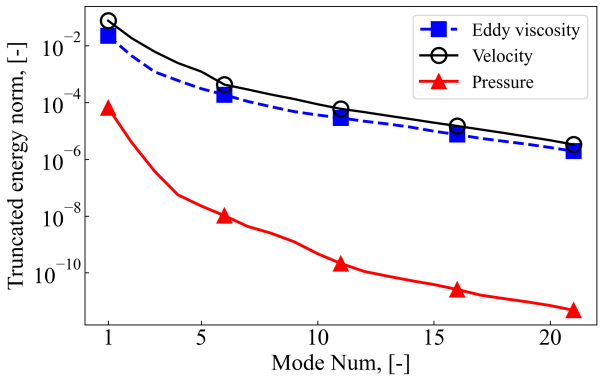


Figure 4.6: Relative truncation energy ($\mathcal{E}^2$, see equation (3.7)) of different numbers of POD modes of velocity, pressure, and turbulent viscosity fields. $N_{\text{train}} = 30$.

mainly represent the effect of inflow conditions at the entrance region. The last one captures the flow development in the middle.

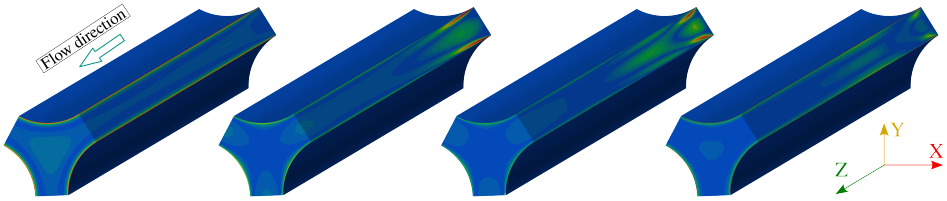


Figure 4.7: Magnitude of the first four POD modes of velocity. From left to right: modes No. 1, 2, 3, and 4. $N_{\text{train}} = 30$.

It should be noted that the lifting functions influence the results of POD. Different strategies to obtain the lifting functions are described in Section 3.2. Additionally, the number of snapshots may affect modes, especially when the training set is small. Generally, more samples yield a better approximation. However, the improvement is not significant when the sample size is large enough.

Once POD basis functions are available, the POD-Galerkin is applied and analyzed first, followed by the PODI technique. Their performance is quantified with the corresponding FOM solutions. For example, the relative L^2 error norm of velocity is computed with $\epsilon_{\mathbf{u}}(\mu_j) = \frac{\|\mathbf{u}(\mu_j) - \mathbf{u}_{\text{ROM}}(\mu_j)\|_{L^2(\Omega)}}{\|\mathbf{u}(\mu_j)\|_{L^2(\Omega)}}, \quad j = 1, \ldots, N_\mu$.

The influence of several factors will be presented for the two types of ROMs. For the intrusive method, the analysis includes the effect of Dirichlet condition treatment and the stabilization techniques. For the non-intrusive technique, three interpolation algorithms are compared. Two common aspects, i.e., the number of POD modes and the size of the training set, are also considered. Lastly, the two frameworks are compared.

4.1.2.2 POD-Galerkin ROMs

Different ROM formulations As mentioned in Section 3.2.2.2, the POD-Galerkin ROM can be formulated in four different ways, depending on the treatments of Dirichlet conditions (i.e., lifting functions and penalty terms) and stabilization techniques (i.e., supremizer enrichment (SUP) and pressure Poisson equation (PPE)).

Now, let us compare the four POG-Galerkin ROM formulations listed in Section 3.2.2.2, i.e., (i) lifting function and supremizer enrichment (*SUP_lift*); (ii) penalty method and supremizer enrichment (*SUP_penalty*); (iii) lifting function and PPE (*PPE_lift*); (iv) penalty method and PPE (*PPE_penalty*). In this case, ROMs are constructed with 30 solutions. Since 20 test samples are selected, the mean relative L^2 error of velocity fields is adopted to assess the ROM accuracy: $\overline{\epsilon_{\mathbf{u}}} = \frac{1}{N_{\text{test}}} \sum_{j=1}^{N_{\text{test}}} \epsilon_{\mathbf{u}}(\mu_j)$. Remark that, without specific indication, the *error* is hereafter denoted $\overline{\epsilon_{\mathbf{u}}}$.

A comparison of the four strategies is shown in Fig. 4.8. The x-axis is the mode number of velocity and eddy viscosity (varies from 2 to 10), while the pressure basis functions are fixed to two. That is because the truncation energy norm of the pressure decays faster than that of the other two fields (see Fig. 4.6). The y-axis denotes the mean error of all test samples. The projection error $\mathcal{E}$ (see equation (3.6)) is also plotted in the figure as a reference, which represents the best accuracy that can be achieved with the current POD basis.

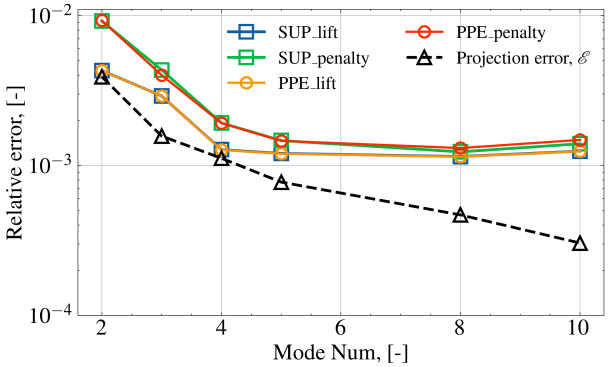


Figure 4.8: Mean relative L^2 error of velocity considering four ROM configurations: (i) lifting function and supremizer enrichment (*SUP_lift*); (ii) penalty method and supremizer enrichment (*SUP_penalty*); (iii) lifting function and PPE (*PPE_lift*); (iv) penalty method and PPE (*PPE_penalty*). The projection error $\mathcal{E}$ is defined in equation (3.6) as a reference value. The sizes of the training and test sets, N_{train} and N_{test}, are 30 and 20, respectively. The lines vary with the number of velocity ($\mathbf{u}$) and eddy viscosity (ν_t) POD modes, and the pressure modes are fixed to 2.

As shown, the lifting approach is slightly better than the penalty method. The two stabilisation procedures yield similar accuracy, as confirmed in a study by Giovanni Stabile et al. [368]. The four lines decrease as the number of modes increases from 2 to 4, and thereafter, the improvement is not significant. Additionally, the errors increase when the number is too large (e.g., 10). This may be due to two reasons: (i) more POD modes can better represent the solution, as indicated by the projection error; (ii) the less relevant high-order modes may affect the solution of the

reduced ODE and reduce the overall accuracy. Thus, the number of POD modes should be determined by considering the decay of the truncation error and by numerical experiments.

In summary, the four formulations can all achieve an acceptable accuracy of 10^{-3} and are close to the projection error. That demonstrates the capability of POD-Galerkin ROMs for the case.

In the following analysis, only one configuration is selected to construct the POD-Galerkin ROMs: *lifting functions* and *supremizer enrichment*, and then the effects of other factors are analyzed. They are chosen due to several reasons. The penalty for Dirichlet conditions requires a proper penalty coefficient, which must be tuned through a series of numerical experiments. That increases the complexity of building ROMs. In contrast, the lifting approach is more straightforward. The stabilization method, the supremizer enrichment, is adopted because it is more frequently used in the references (see the explanation in Section 3.2.2.1).

Effect of the size of training set In general, the number of training samples is decided by an independent parameterization study that specifies the dimension of the parameter space and the parameter ranges. The number of snapshots can affect the POD modes, and thus the performance of ROMs. Although parameterization is not considered here, the influence of different N_{train} values (5, 10, 20, and 30) is analyzed. The same test set ($N_{\text{test}} = 20$) is used to validate the ROMs. The samples are shown in Figs. D.1 and D.2. Similarly, considering the decay of POD truncation energy norm, the number of velocity and eddy viscosity modes varies from 2 to 10, and the pressure modes are fixed to 2.

The comparison is presented in Fig. 4.9. It can be seen that the accuracy generally improves as N_{train} increases from 5 to 10. The improvement is not significant when N_{train} changes from 10 to 30. This indicates that 10 samples are already sufficient to capture the main features of the parametric problem.

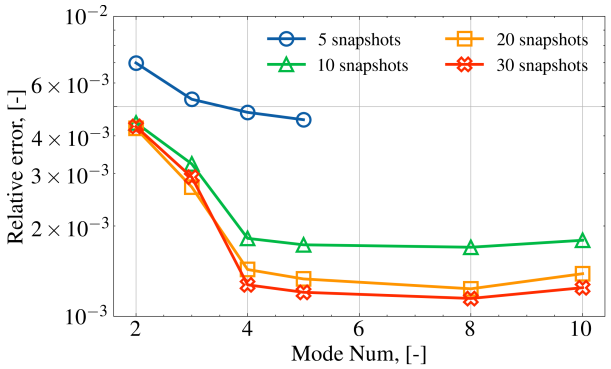


Figure 4.9: Mean relative L^2 error of velocity (a ROM utilizing lifting functions and the supremizer) with different sizes of training set, i.e., $N_{\text{train}} = 5, 10, 20, 30$. The same test samples $N_{\text{test}} = 20$ are used to quantify the ROMs. The lines vary with the number of $\mathbf{u}$ and ν_t POD modes, and the pressure modes are fixed to 2.

Additionally, except for the case with five training samples, the variation trends of the other three lines are similar. The error decreases as the number of modes increases from 2 to 4, and then

remains nearly unchanged. The results are consistent with the previous analysis (see Fig. 4.8). That highlights the importance of numerical tests to determine the mode number.

4.1.2.3 PODI ROMs

The focus for PODI ROMs is on the impact of algorithms on the regression of parameters and POD coefficients. As explained in Section 3.2.3, RBF, GPR, and ANN are involved and compared.

Note that the same dataset setting is employed for the three algorithms without specific clarifications. The size of training and test sets is fixed to $N_{\text{train}} = 30$ and $N_{\text{test}} = 20$, respectively.

Radial basis function (RBF) The analysis begins with RBF. Several widely used kernels are considered, including *Linear*, *Thin-plate spline*, *Gaussian*, *Cubic*, and *Quintic* (see Appendix B.3.1 for their formulations).

The plot shows that, except for the *linear* kernel, the other four kernels can achieve similar accuracy of about 10^{-3}. The *Cubic* and *Quintic* are a little better than the other two. An interesting phenomenon is that the error decreases when $N_{\mathbf{u}}^{\text{RB}}$ increases from 2 to 3, and then remains almost unchanged from 3 to 10. That trend is also observed in POD-Galerkin results (see Figs. 4.8 and 4.9). It reveals that the benefit of more modes is offset by regression errors for high-order POD coefficients.

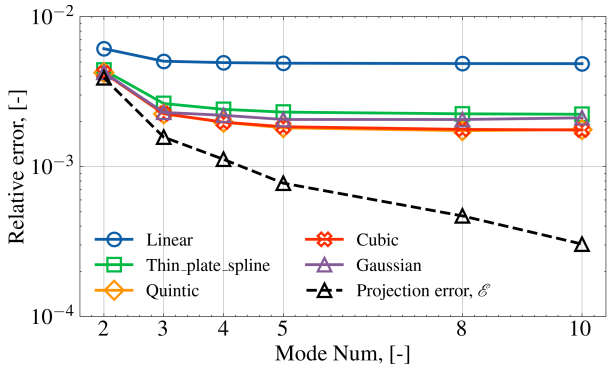
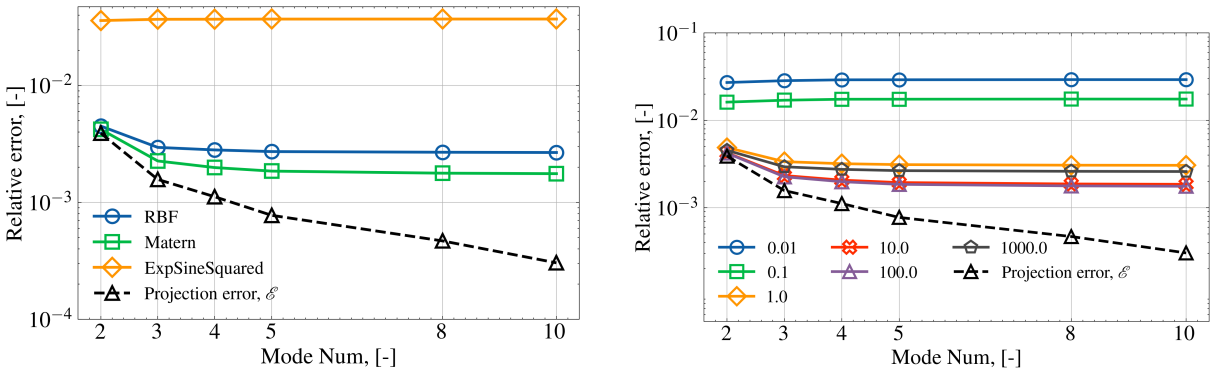


Figure 4.10: Mean relative L^2 error of velocity (a PODI ROM with RBF) against the number of $\mathbf{u}$ and ν_t POD modes. Five kernels are compared, whose formulations are given in Section 3.2.3.1. $N_{\text{train}} = 30$ and $N_{\text{test}} = 20$.

Gaussian Process regression (GPR) The next approach is GPR, and three kernels are compared, i.e., *Squared Exponential*, *Matern*, and *RBF* (see definitions in Appendix B.3.2). The best approximation is given by the *Matern* kernel, as shown in Fig. 4.11a. The variances of the three lines are similar to the previous RBF results. The error decreases as $N_{\mathbf{u}}^{\text{RB}}$ increases from 2 to 3, and then it is nearly unchanged.



(a) Different kernels

(b) Length scales for *Matern*

Figure 4.11: Mean relative L^2 error of velocity (a PODI ROM with GPR) against the number of $\mathbf{u}$ and ν_t modes. $N_{\text{train}} = 30$ and $N_{\text{test}} = 20$. (a) Effects of three kernels, whose formulations are given in Section 3.2.3.2; (b) Influence of length scales in the *Matern* kernel.

The GPR kernels include hyperparameters that can affect prediction accuracy. The length scale (l) is generally regarded as the most important one. It scales the distance between two points in the parameter space. Fig. 4.11b displays the accuracy when using different l for the *Matern* kernel. The best approximations are $l = 10$ and 100. A value that is too small or too large can lead to inaccuracy.

It can be seen that a poor choice of kernel and length scale can lead to very poor performance. Therefore, a test procedure is necessary when using the machine algorithms.

Artificial neural networks (ANN) The last technique is ANN. The prediction of an ANN depends on various factors, including the activation function, the number of hidden layers, and the number of neurons per layer. The influence of the three aspects is analyzed below. Other items, e.g., the optimization algorithm and the number of training epochs, might also affect the results, but they are not the focus of this research and thus fixed (see Table D.1).

Fig. 4.12 shows the performance of different activation functions, i.e., Rectified Linear Unit (*ReLU*), *Tanh*, and *Sigmoid* (see their definition in Section 3.2.3.3). The worst predictions are obtained by *ReLU*. In contrast, the last two can provide similar accuracy. For the mode numbers from 2 to 5, the values are very close to the projection error. That indicates the ANN can almost reach the best accuracy of the current POD basis in the range. Another point is that for x values equal to 8 and 10, *Tanh* and *Sigmoid* are nearly stable. The trend is common for all three interpolation techniques.

The architecture of the ANN is another crucial aspect. Considering the results above, the activation function is fixed to *Tanh*. The number of hidden layers and the number of neurons per layer are two aspects to be analyzed. Seven configurations listed in Table 4.3 are compared, with 2, 3, or 4 hidden layers and neurons per layer of 3, 5, 10, 20, and 30.

The results are presented in Fig. 4.13. The left plot shows the effects of hidden layer size (2, 3, and 4). The differences are not significant.

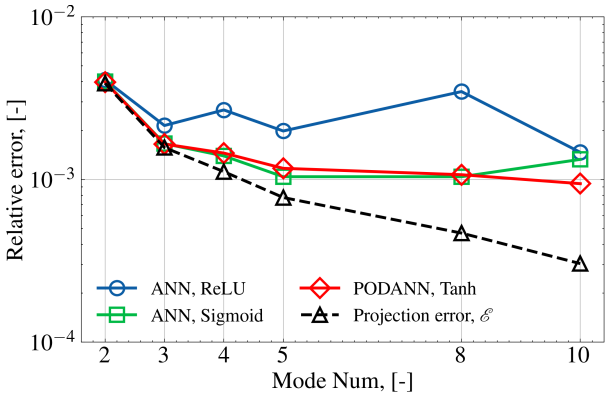


Figure 4.12: Mean relative L^2 error of velocity (a PODI ROM with ANN) against the number of $\mathbf{u}$ and ν_t modes. Three activation functions are compared, i.e., *Tanh*, *ReLU*, and *Sigmoid*. $N_{\text{train}} = 30$ and $N_{\text{test}} = 20$. The number of hidden layers and the number of neurons per layer are fixed at 3 and 10, respectively.

Table 4.3: Architectures of the ANN. The activation function is *Tanh*.

No.	Hidden layers	Neurons per layer
1	2	10
2	3	10
3	4	10
4	3	3
5	3	5
6	3	20
7	3	30

Figure 4.13 (right) displays the influence of different numbers of neurons per layer (3, 5, 10, 20, and 30). It reveals that a too-simple network structure (e.g., three neurons per layer) can lead to inaccuracies for high-order modes. The line for five neurons leads to an increase in the error for $N_{\mathbf{u}}^{\text{RB}} = 8$ and 10. The remaining three show similar decreasing trends, but it can still be observed that more neurons can slightly improve ROM performance. However, the increase in neurons will lead to more training time. Thus, a trade-off should be made in practice.

Based on the above analysis, the architecture with three hidden layers and 10 neurons per layer is a reasonable choice, which can provide acceptable accuracy at minimal training time.

Train set size Now, let us evaluate the three interpolation methods together. The best configurations of each algorithm are selected to construct PODI ROMs and compared as follows. For RBF, the *Cubic* or *Quintic* kernel is employed. For GPR, the *Matern* kernel with length scale $l = 100$ is chosen. For ANN, the *Tanh* activation function with three hidden layers and 10 neurons per layer is applied.

Fig. 4.14 compares their results against the number of velocity modes. Four sizes of training set are considered, i.e., $N_{\text{train}} = 5, 10, 20, 30$. The same test set ($N_{\text{test}} = 20$) is used for the

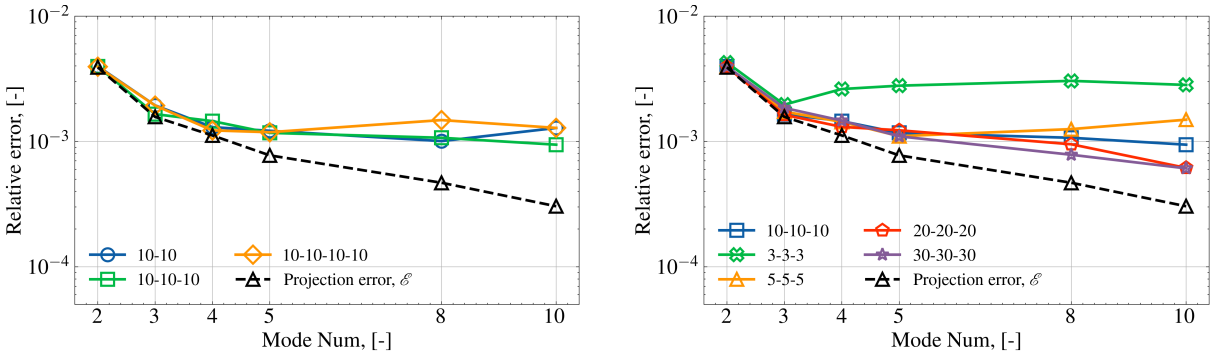


Figure 4.13: Mean relative L^2 error of velocity (a PODI ROM with ANN) against the number of $\mathbf{u}$ and ν_t modes. $N_{\text{train}} = 30$ and $N_{\text{test}} = 20$. Left: effects of different numbers of hidden layers (2, 3, and 4); Right: influence of different numbers of neurons per layer (3, 5, 10, 20, and 30). The activation function is *Tanh*.

validation. It shows that the ANN performs worst on five samples and best on 30 samples. That illustrates that ANNs require more data to train, which is consistent with common knowledge of neural networks. RBF and GPR can provide similar accuracy and variation trends for $N_{\text{train}} = 5, 20,$ and 30.

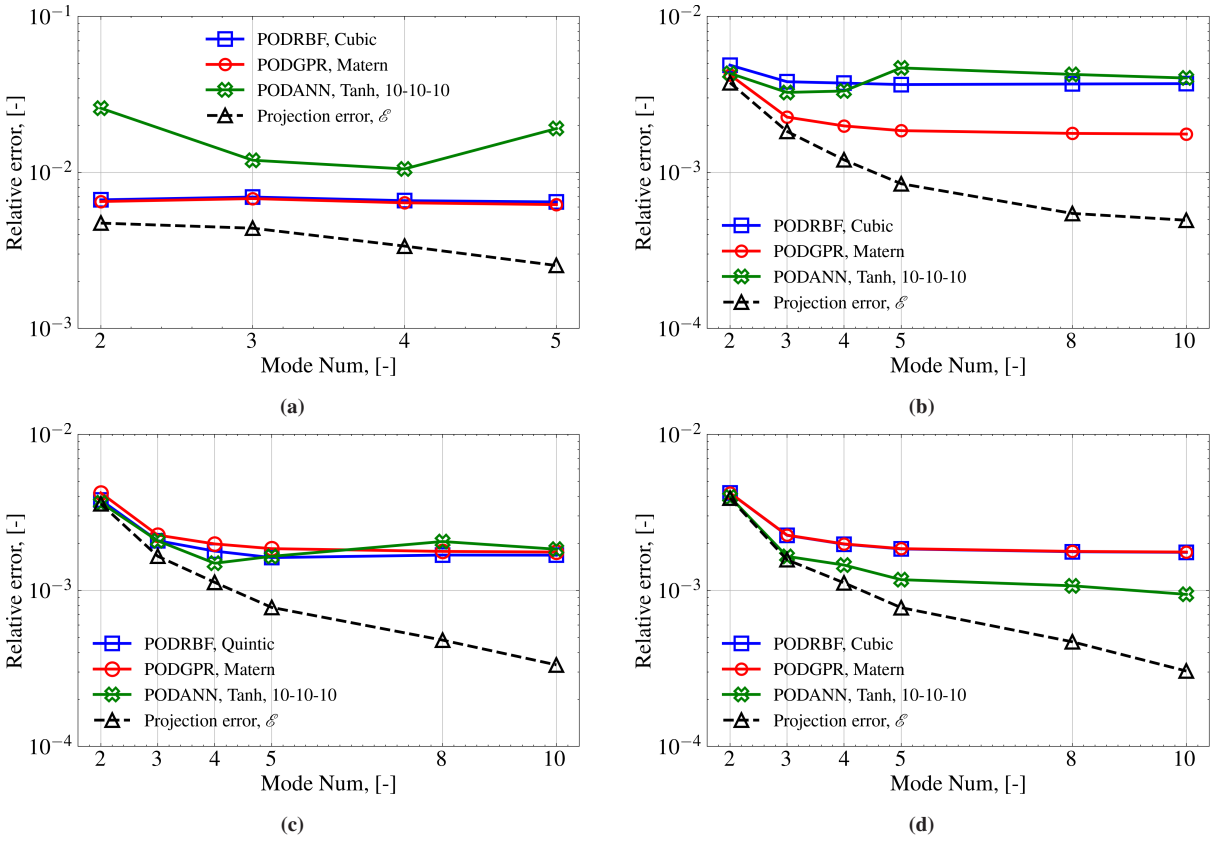


Figure 4.14: Mean relative L^2 error of velocity of PODI ROMs with RBF, GPR, and ANN against the number of $\mathbf{u}$ and ν_t modes. (a) $N_{\text{train}} = 5$; (b) $N_{\text{train}} = 10$; (c) $N_{\text{train}} = 20$; (d) $N_{\text{train}} = 30$. $N_{\text{test}} = 20$ is used to quantify the ROMs.

The previous plots already indicate how the size of the training set affects the performance of different PODI ROMs. The results for RBF and ANN are shown in Fig. 4.15. N_{train} is set to 5,

10, 20, and 30, and $N_{\text{test}} = 20$. A common tendency observed in both figures is that accuracy increases with larger N_{train}. For RBF, the improvement from 5 to 20 samples is noticeable, while the two lines for $N_{\text{train}} = 20$ and 30 are very close, which is also observed in POD-Galerkin ROMs (see Fig. 4.9). This indicates that $N_{\text{train}} = 20$ is already sufficient. For ANN, performance improves continuously, especially for high-order modes.

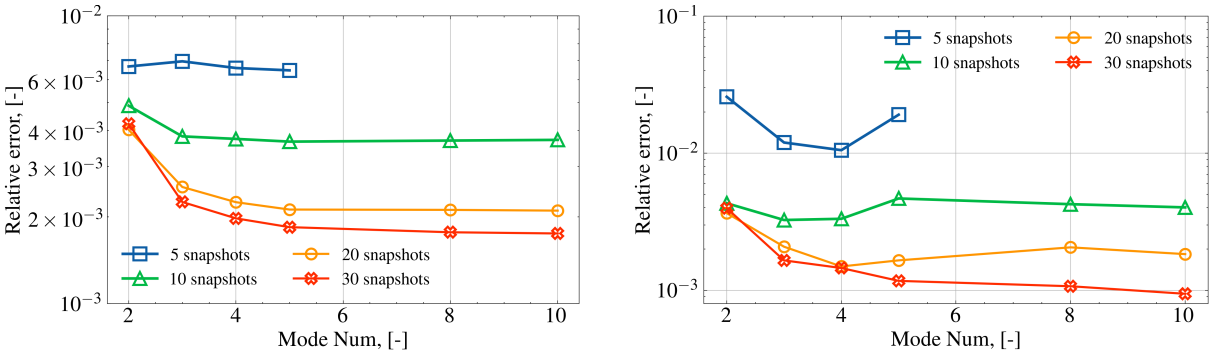


Figure 4.15: Mean relative L^2 error of velocity of PODI ROMs with different sizes of training set against the number of $\mathbf{u}$ and ν_t modes. Left: RBF with *Cubic* kernel; Right: ANN with *Tanh* and $10-10-10$ hidden layers. The standard test set is $N_{\text{test}} = 20$.

4.1.2.4 Comparison and discussion

Now, it is suitable to compare the reduced basis method and PODI techniques. Fig. 4.16 shows variances of POD-Galerkin and PODI (RBF and ANN) ROMs against the number of velocity modes. To improve the readability, only three sizes of training set are included, i.e., $N_{\text{train}} = 5, 10,$ and 30. They are validated with the same high-fidelity solutions, $N_{\text{test}} = 20$.

The plots show that the intrusive methods are much better than the non-intrusive ones when $N_{\text{train}} = 5$ and 10. The value of POD-Galerkin ROMs with 10 samples is very close to that of PODI-RBF with 30 samples. For $N_{\text{train}} = 30$, PODI-ANN is the best.

The results indicate that POD-Galerkin ROMs are more reliable when the number of snapshots is insufficient. The accuracy of PODI ROMs is gradually improved as N_{train} increases. Moreover, PODI methods perform better with more training data. It should also be emphasized that the ANN can achieve higher accuracy with a sufficiently large training set.

As indicated earlier, ROM is primarily used to reduce computational costs for parametric problems. Thus, let us compare the computational costs of the ROMs. Both FOM and ROM calculations are performed on a desktop with an Intel Core i7-10700 CPU (8 cores, 2.9 GHz). The simulation times are listed in Table 4.4. The average FOM time is around 3850 seconds per simulation. $N_{\text{train}} = 30$ and $N_{\mathbf{u}}^{\text{RB}} = 20$ are used for constructing ROMs.

The offline stage of POD-Galerkin ROMs includes POD, supremizer solutions, and projection operation (i.e., construction of ROM algebraic systems). The reduced ODE is solved in the

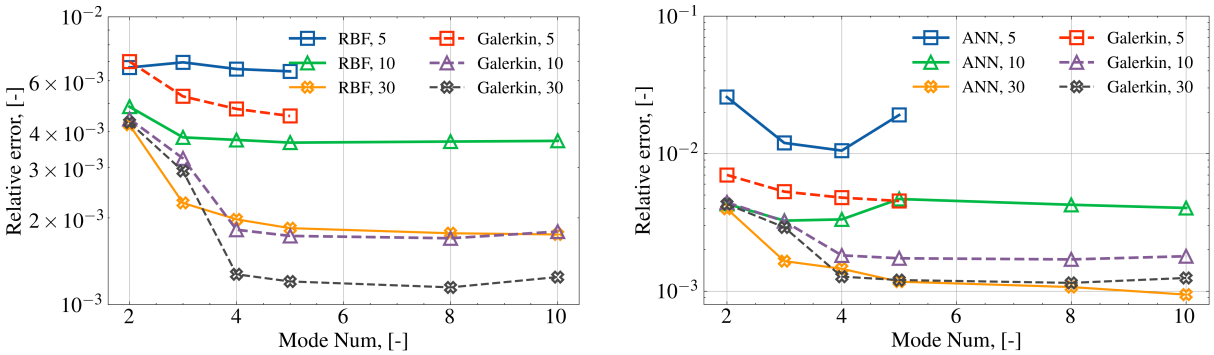


Figure 4.16: Mean relative L^2 error of velocity of POD-Galerkin and PODI (i.e., RBF and ANN) ROMs against the number of $\mathbf{u}$ and ν_t modes. Left: RBF and POD-Galerkin; Right: ANN and POD-Galerkin. $N_{\text{train}} = 5, 10,$ and 30 and $N_{\text{test}} = 20$.

online stage. The projection is the most time-consuming, but it is still 100 times less than FOM. Then, the cost for solving a reduced ODE is negligible.

For PODI ROMs, POD is performed first. Then, various interpolants can be built to fit parameters and POD coefficients. It can be seen that ANN training is comparatively expensive, since it must pass through all hidden layers. The prediction of PODI ROMs is very fast, especially for RBF and GPR. The ANN is slightly slower at prediction due to its complexity.

It can be seen that the POD cost differs between the two types of ROMs. That is because different libraries are used to compute the POD. For the POD-Galerkin method, the POD is computed using *Eigen* [387], a high-performance C++ linear algebra library. While for PODI, *NumPy* [369] is employed. The computational time is hence different.

In short, the acceleration ratio is around 10^4 to 10^5 when only the solving time is considered. Even including the POD and ROM construction, the speedup is still more than 10^2 and up to 10^3.

Lastly, it should be noted that the POD-Galerkin ROM can approximate all the fields (i.e., velocity, pressure, and eddy viscosity) simultaneously. In contrast, PODI ROMs must build separate models for each field. The table only lists the time of velocity for PODI.

Note that the computational cost of ROMs depends on various factors, such as the number of modes, the size of the training set, and the interpolation algorithm. The duration for file *I/O* is nonnegligible. As each ROM calculation is very fast, the time might be affected by other processes of the computer. In short, the values in Table 4.4 are only for reference.

According to the above comparison, both the POD-Galerkin and the PODI-ROM methods can provide an acceptable approximation for the current case. The mean relative L^2 error norm of velocity is around the order of 10^{-3}.

Table 4.4: Computational time of FOM and ROM calculations with $N_{\text{train}} = 30$ and $N_{\text{test}} = 20$. The offline and online stages are listed separately. The FOM time is per simulation. The ROM time is the sum of values for 20 test samples. The POD-Galerkin ROM employs lifting functions and the supremizer. The PODI-RBF uses the *Cubic* kernel, the PODI-GPR adopts *Matern* kernel, and the PODI-ANN has three hidden layers and 10 neurons per layer with *Tanh* activation function.

	Item	FOM	Galerkin	RBF	GPR	ANN
	FOM (per)	3850		-		
	POD	-	0.2		1.5	
Offline (s)	Supremizer	-	5.4		-	
	Projection	-	24.0		-	
	Interpolation	-	-	0.3	0.3	8.5
Online (s)	Solving ($N_{\text{test}} = 20$)	-	2.0e-2	5.0e-3	5.0e-3	0.5

4.2 A seven-pin bare rod bundle

4.2.1 High fidelity simulations

The second geometry is a seven-pin bare rod bundle. Note that most of the numerical configurations are already illustrated in the case above and are still valid for this problem. Only the necessary CFD setups and results are clarified here, along with a few results.

The 3D model of the bundle is shown in Fig. 4.17. Dirichlet inflow and Neumann outflow conditions are imposed on the two sides of the bundle. All the walls are regarded as non-slip. The bundle length is chosen as $L = 200$ mm to ensure a fully developed outflow. Note that the sizes of the domain are the same as the subchannel, except for an additional length of bundle edge $E = 17.97$ mm.

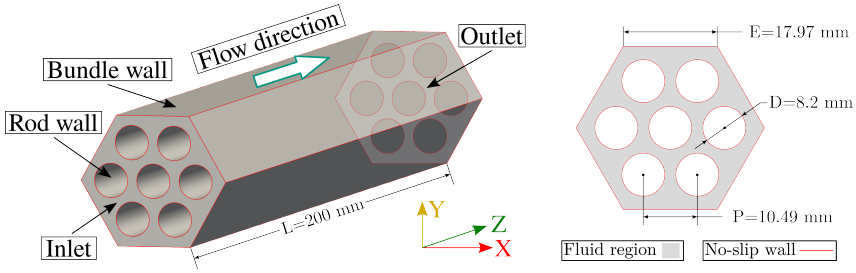


Figure 4.17: Seven-pin bare rod bundle. Left: geometry. Right: boundary conditions

The range of the three parameters (see Table 4.2) is the same as in the previous case. The uniform and power-law inlet profiles are shown in Fig. D.3. The size of the parameter set P is also 50, but in this case, they are selected in two steps:

- Uniform grid points ($\tilde{N}_\mu$ points) within the ranges are chosen, denotes P, as shown in Fig. D.4.

- An *iterative* approach, Algorithm 4, is applied to expand P with points from a candidate set P_c. Note that P_c is created considering a uniform random distribution.

The final samples are plotted in Fig. D.6.

The structured grid of the bundle is displayed in Fig. 4.18, with a total number of around 5.1 million cells. The mesh is generated to satisfy the requirements of a SST $k-\omega$ RANS simulation. Several aspects are considered, such as y^+, orthogonality, skewness, etc.

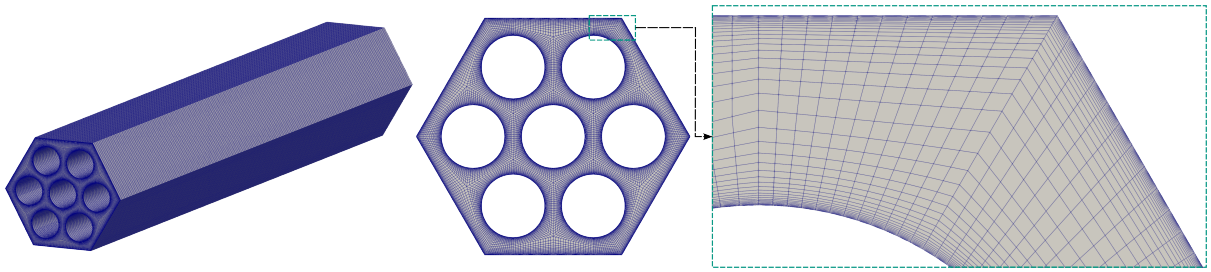


Figure 4.18: Structured mesh of the bare rod bundle.

The numeric schemes of the FOM calculations are listed in Tables C.1 and C.2. Two velocity fields are presented in Fig. 4.19. To better show the distribution inside, a half-view of the bundle is shown. One can easily notice the development of the flow profiles. More results will be given in the ROM analysis.

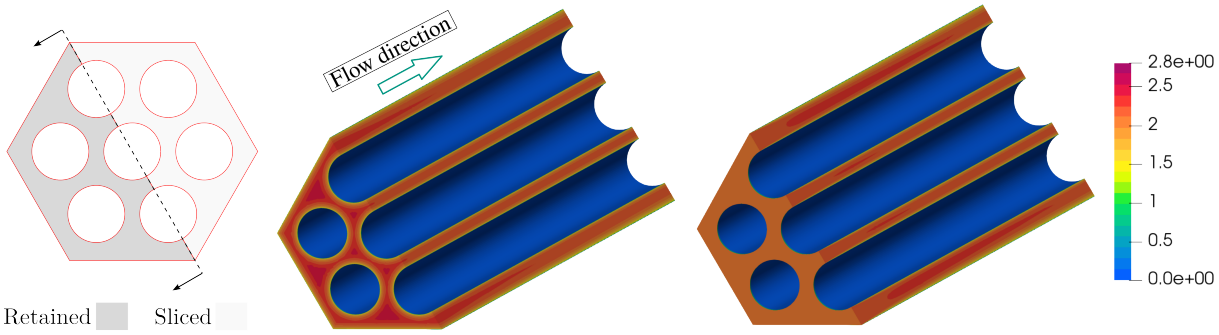


Figure 4.19: Velocity Contours. Left: the cut plane. Middle: power-law inlet, $\mu_1^L = 0.0$, $\mu_2^L = 2.1$, and $\nu = 2.327 \times 10^{-7}$ m^2/s. Right: uniform inflow, $\mu_1^L = 2.1$, $\mu_2^L = 0.0$, and $\nu = 2.327 \times 10^{-7}$ m^2/s.

4.2.2 Reduced order models

The results for the seven-pin bare rod bundle are analyzed in the same manner as those above. The POD modes are visualized first. The performance of POD-Galerkin and PODI ROMs is then evaluated and compared with $N_{\text{train}} = 5, 10, 20,$ and 30 and $N_{\text{test}} = 20$ (plotted in Fig. D.6). In most scenarios, the same settings as in the previous case are employed, unless otherwise specified.

4.2.2.1 POD and dominant modes

The POD truncation energy ($\mathcal{E}^2$, see equation (3.7)) of velocity, pressure, and eddy viscosity against the number of modes is shown in Fig. 4.20. Here, the reduction is performed on $N_{\text{train}} = 30$. Before performing POD, the velocity fields are homogenized with lifting functions (i.e., the solutions with $\mu_1^L = 1, \mu_2^L = 0$ and $\mu_1^L = 0, \mu_2^L = 1$).

Fig. 4.20 reveals that the variances of the three lines are exponential, which is similar to the previous case. If the tolerance is set to 10^{-4}, five velocity and eddy viscosity modes are required, whereas the value is one for pressure.

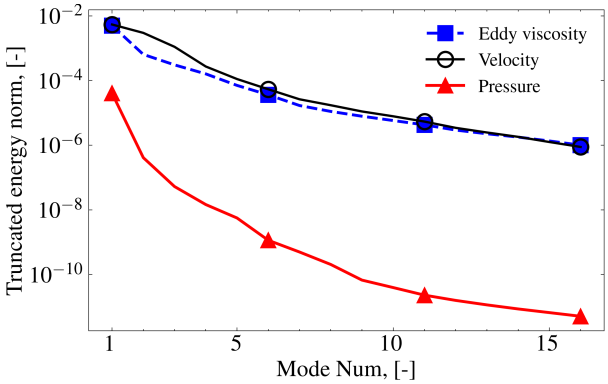


Figure 4.20: POD relative truncation energy ($\mathcal{E}^2$, see equation (3.7)) of velocity, pressure, and eddy viscosity against the number of modes for the seven-pin bare rod bundle. $N_{\text{train}} = 30$.

The first four dominant flow patterns (velocity magnitude) are visualized in Fig. 4.21. It can be seen that the features in the center subchannel are similar to those of the subchannel case (see Fig. 4.7). Since the overall bulk flow is involved in the lifting functions, the first mode mainly captures the near-wall flow. The others represent the development of flow due to parameterized inflow conditions in the first half of the channel.

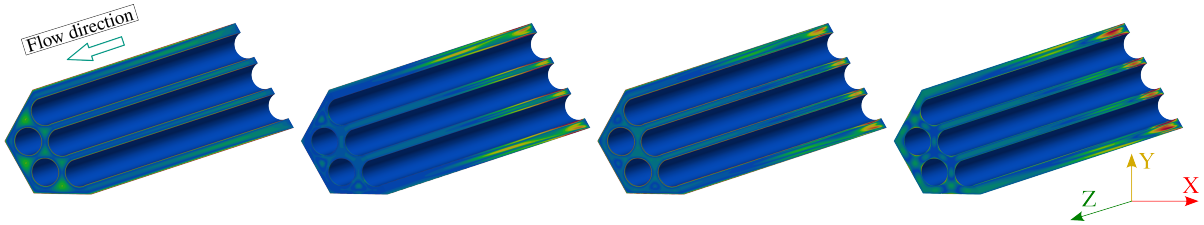


Figure 4.21: First four dominant features of velocity magnitude for the bare rod bundle. $N_{\text{train}} = 30$. From left to right: modes No. 1, 2, 3, and 4.

4.2.2.2 POD-Galerkin ROMs

For the POD-Galerkin ROMs, the focus is on the influence of the training set size. The lifting functions and the supremizer are still employed for the construction.

The error of velocity against the number of velocity modes with different sizes of training set is shown in Fig. 4.22. $N_{\text{train}} = 5, 10, 20,$ and 30 and a common validation set ($N_{\text{test}} = 20$) is used. The sampling is plotted in D.6. The number of pressure modes is fixed to 2, while that of the velocity and eddy viscosity modes varies from 2 to 10.

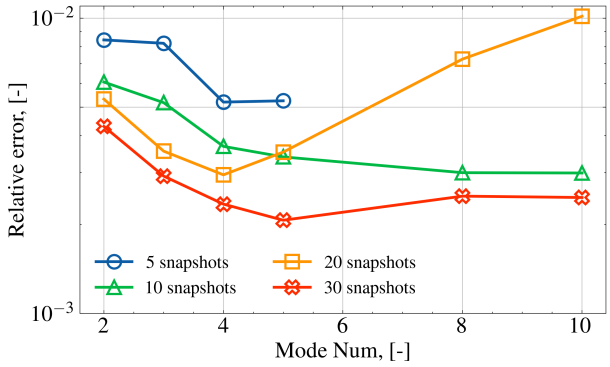


Figure 4.22: Mean relative L^2 error of velocity (POD-Galerkin based ROM utilizing lifting functions and the supremizer enrichment) with different sizes of training set, i.e., $N_{\text{train}} = 5, 10, 20, 30$. The same test samples $N_{\text{test}} = 20$ are used to quantify the ROMs. The lines vary with the number of $\mathbf{u}$ and ν_t POD modes, and the pressure modes are fixed to 2.

The overall trend is that the accuracy improves as N_{train} increases. Besides, four velocity and eddy viscosity modes are appropriate values to model the problem. The error is around 2×10^{-3}, which is still good for engineering applications.

A special variance is that the error with $N_{\text{train}} = 20$ gradually increases for $N_{\mathbf{u}}^{\text{RB}}$ greater than 5. That is because four cases with the minimum inflow velocity are poorly predicted for more than four modes, with an error of around 0.05. Thus, the averaged value is highly deteriorated. Maybe the modes extracted from 20 samples focus on other regions of the parameter space with higher flow rates, leading to inaccuracy in these four cases. The results indicate that the POD-Galerkin method does not guarantee robust performance, and validation is crucial.

4.2.2.3 PODI ROMs

The PODI ROMs with RBF, GPR, and ANN are evaluated in the following paragraphs. Firstly, let us discuss how the size of the training set affects the performance of PODI ROMs. Fig. 4.23 shows the results of POD-RBF and POD-GPR ROMs with $N_{\text{train}} = 5, 10, 20,$ and 30. They are quantified with the same test set ($N_{\text{test}} = 20$). Note that the best kernels are employed to build surrogates, i.e., *Cubic* or *Quintic* for RBF and *RBF* or *Matern* for GPR. The number of pressure modes is fixed to 2, while those of velocity and eddy viscosity modes vary from 2 to 10. Note that the POD configurations are determined by comparisons shown in Figs. D.7 and D.8.

It is clear from the lines that accuracy improves as N_{train} increases. Another interesting observation is that all the lines for $N_{\text{train}} = 5$ and 10 are almost overlapped, which is different from

the POD-Galerkin ROMs. That indicates the PODI ROMs with RBF and GPR are not practical when the data is insufficient. The accuracy is improved as N_{train} increases to 20 and 30.

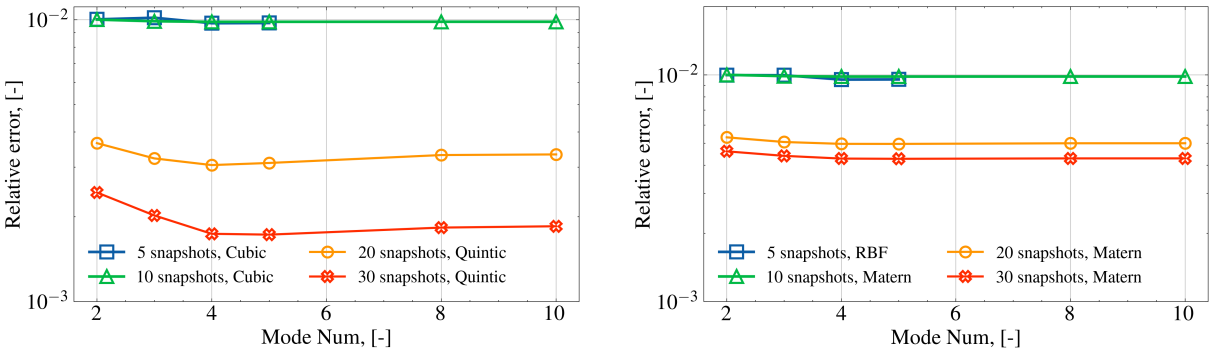


Figure 4.23: Mean relative L^2 error of velocity of PODI ROMs with $N_{\text{train}} = 5, 10, 20,$ and 30 against the number of $\mathbf{u}$ and ν_t modes. Left: RBF with *Cubic* or *Quintic* kernel; Right: GPR with *RBF* or *Matern* kernel. Quantified by the common test set $N_{\text{test}} = 20$.

The effects of training set size on PODI-ANN ROMs are illustrated in Fig. 4.24. The architecture is fixed to 3 hidden layers with 10 neurons per layer, using the *Tanh* activation function. Interested readers can refer to Fig. D.9 for the influence of network structure and activation functions, respectively.

The results show that accuracy improves significantly as N_{train} increases. The line for five samples is in the order of 10^{-1}, while that for 30 samples is around 10^{-3}. When $N_{\text{train}} = 20$, the approximations for low-order modes are acceptable, but the error increases for high-order modes. That might be improved when employing a more complex network structure.

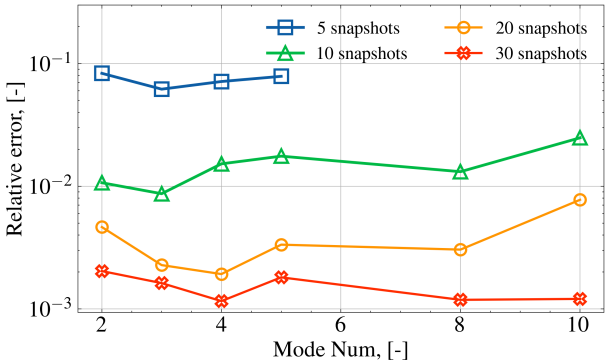
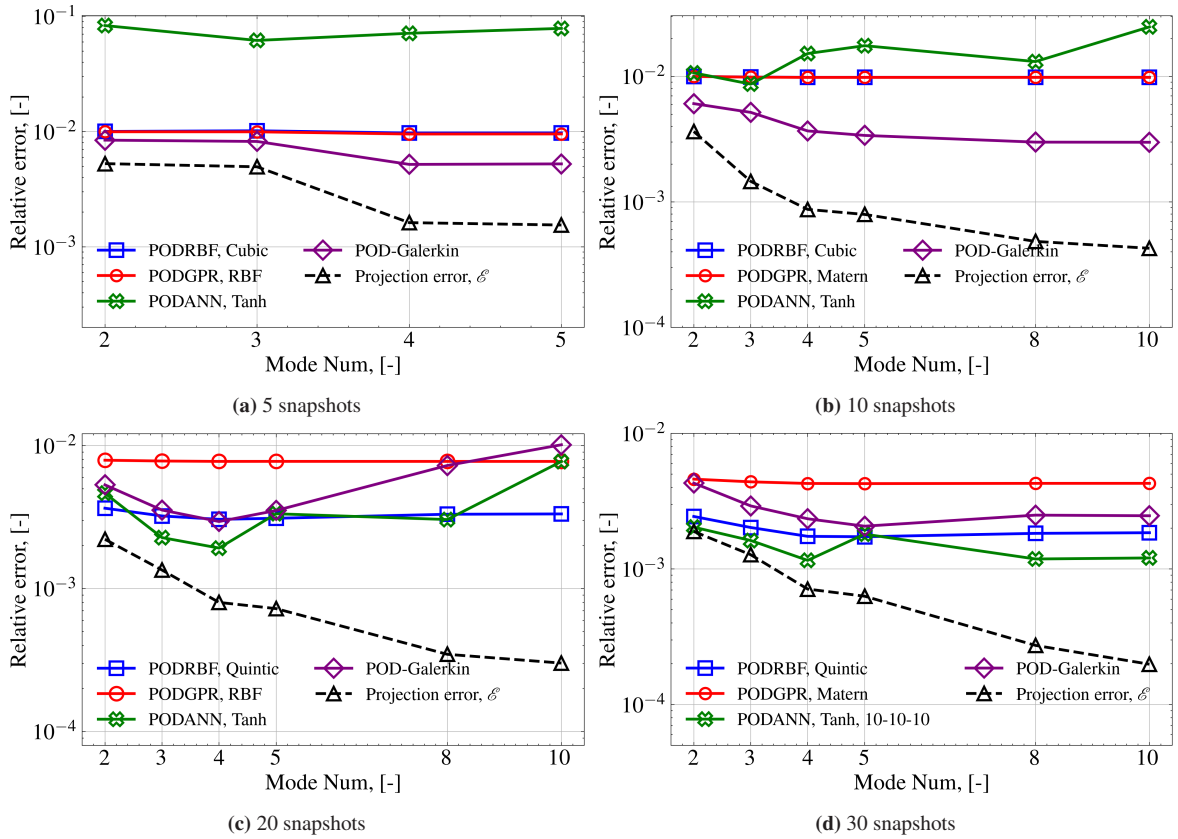


Figure 4.24: Mean relative L^2 error of velocity of PODI ROMs with $N_{\text{train}} = 5, 10, 20,$ and 30 against the number of $\mathbf{u}$ and ν_t modes. The ANN is configured with *Tanh* activation function and $10 - 10 - 10$ hidden layers. Test set is $N_{\text{test}} = 20$.

4.2.2.4 Comparisons and discussions

The last step is an overall evaluation of all ROMs. Fig. 4.25 presents the comparisons of POD-Galerkin and PODI ROMs with different N_{train}. The projection error $\mathcal{E}$, the best approximation

that can be achieved with the current POD basis, is also included for reference. Only results for velocity fields are shown.



(a) 5 snapshots

(b) 10 snapshots

(c) 20 snapshots

(d) 30 snapshots

Figure 4.25: Comparisons of POD-Galerkin and PODI ROMs for the bare rod bundle case with different sizes of training set. The projection error is also included for reference. The results are obtained with $N_{\text{train}} = 5, 10, 20, 30$, and validated with a standard test set $N_{\text{test}} = 20$.

The plots are obtained with $N_{\text{train}} = 5, 10, 20, 30$, and validated with a standard test set $N_{\text{test}} = 20$. A common observation is that performance improves as N_{train} increases. POD-Galerkin ROMs are more accurate for $N_{\text{train}} = 5$ and 10, indicated in Figs. 4.25a and 4.25b. ANN achieves the best predictions when $N_{\text{train}} = 30$ (see Fig. 4.25d), and especially, the error is very close to the projection error for $N_{\mathbf{u}}^{\text{RB}} = 2, 3,$ and 4.

The high-fidelity CFD and ROMs are performed on a cluster using an Intel Xeon Platinum 8358 CPU (64 cores, 2.6 GHz). The computation cost is summarized in Table D.2. The speedup ratio is similar to the previous case, i.e., around 10^4 to 10^5 when only the solving time is considered. The quantities are in the range of 10^2 to 10^3 when including POD and ROM construction (i.e., projection for POD-Galerkin and training for PODI).

4.3 A complex case: a typical wire-wrapped rod bundle

4.3.1 High fidelity simulations

The necessary settings regarding the FOM computation in a seven-pin wire-wrapped bundle will be provided. The description already given above will not be duplicated here.

The 3D model is displayed in Fig. 4.26. The only difference from the bare rod bundle is the existence of wires. A wire (diameter $D_w = 2.2$ mm) is wrapped helically around each fuel rod. The axial distance for one full spiral turn of a wire is called *wire pitch* H_w, which is also adopted as the length of the computational domain $L = H_w = 328$ mm.

As shown in Fig. 4.26, the problem is bounded by a Dirichlet inlet, a Neumann outlet, and no-slip walls. As aforementioned, the viscosity and inflow profiles are parameters.

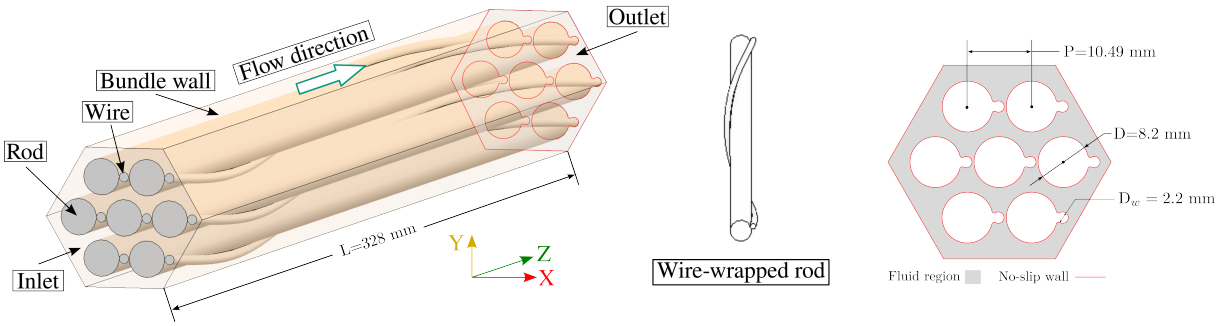


Figure 4.26: Structured mesh of the wire-wrapped rod bundle.

Fig. 4.27 displays the unstructured grid for the wire-wrapped bundle. Narrow gaps exist between the wire and the rods, which is a critical and complex part for CFD simulation. Thus, very thin boundary cells are used to capture the flow distribution. Due to the complexity of the wires, it is challenging to discretize the domain using a structured grid. Hence, an unstructured mesh is generated. Besides, SST $k-\omega$ is applied for turbulence modeling, which requires $y^+ \approx 1$ near the walls. Other characteristics of the grid, e.g., quality, aspect ratio, etc, should also be satisfied. Consequently, to well present the requirements, a mesh with more than 36 million cells is generated.

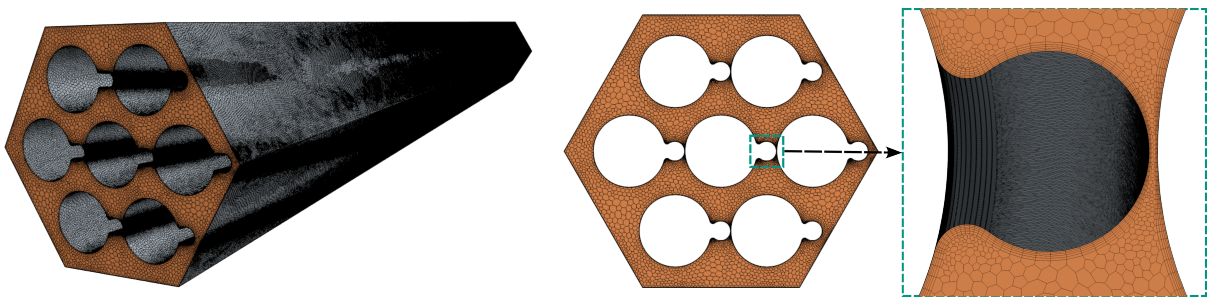


Figure 4.27: Unstructured mesh of the wire-wrapped bundle.

The half-view of velocity contours is given in Fig. 4.28. The helical wire-wrap design induces complex flow patterns, including strong secondary flow, low-velocity regions, and enhanced turbulence. Also, it promotes transverse mixing. It can be seen that inlet effects become negligible a short distance from the channel's entrance, because the flow rapidly develops due to the wires. As a result, in the two plots, the velocity features are similar after the second half of the channel.

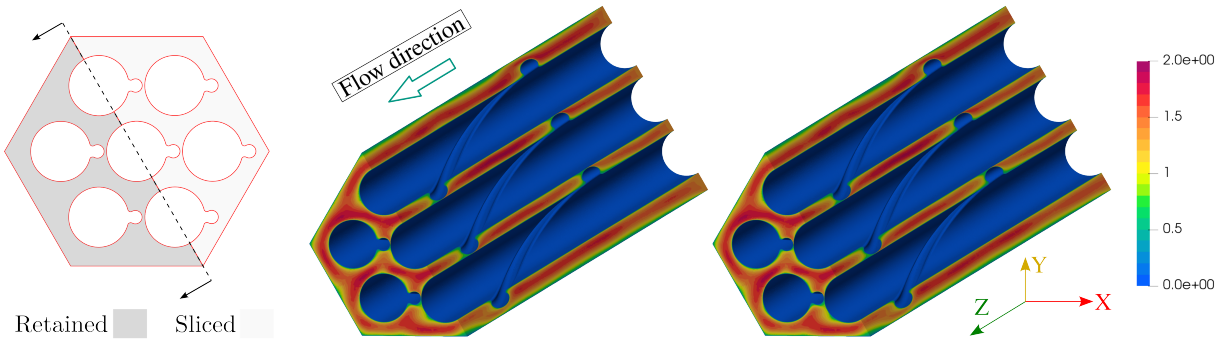


Figure 4.28: Velocity Contours. Left: the cut plane. Middle: power-law inlet, $\mu_1^L = 0.0$, $\mu_2^L = 1.35$, and $\nu = 1.486 \times 10^{-7}$ m^2/s. Right: uniform inflow, $\mu_1^L = 1.35$, $\mu_2^L = 0.0$, and $\nu = 1.486 \times 10^{-7}$ m^2/s.

The streamlines of velocity can better illustrate the complex flow patterns in the wire-wrapped bundle. Fig. 4.29 shows streamlines of velocity in the fluid domain. The plot displays the swirl flows guided by wires. Actually, the wires are specifically designed to increase transverse flow and turbulence, thereby enhancing heat transfer between the rods and the fluid. The strong secondary flows inside the bundle are highly three-dimensional and complex.

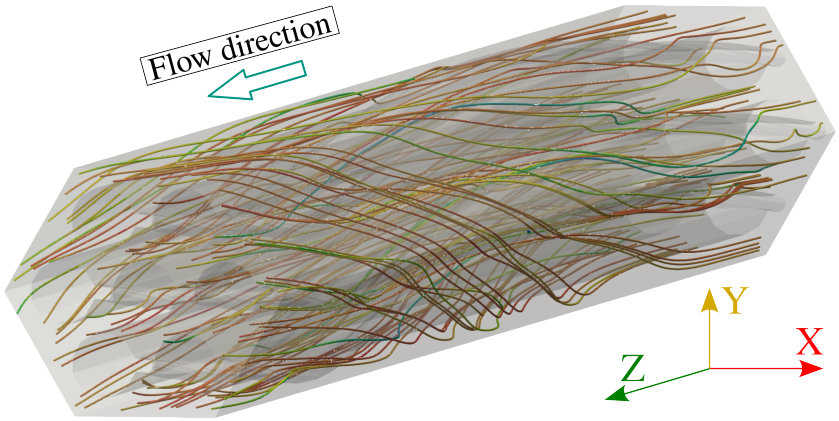


Figure 4.29: Streamlines of velocity colored by velocity magnitude. The fluid flows helically around the rods due to the existence of wires.

4.3.2 Reduced order models

The analysis of ROMs for the wire-wrapped bundle is organized similarly to above, including POD modes, POD-Galerkin ROMs, PODI ROMs, and comparisons.

4.3.2.1 POD and dominant modes

The decay of POD truncation energy ($\mathcal{E}^2$, see equation (3.7)) of velocity, pressure, and eddy viscosity against the number of modes is shown in Fig. 4.30. The pressure line drops very quickly, indicating the first mode can already result in a truncation energy error of about 10^{-4}. For velocity and eddy viscosity, around 10 modes are needed to reach a tolerance of 10^{-4}. Even though the wires enhance the turbulence, a few dominant flow features can still be extracted from the solutions.

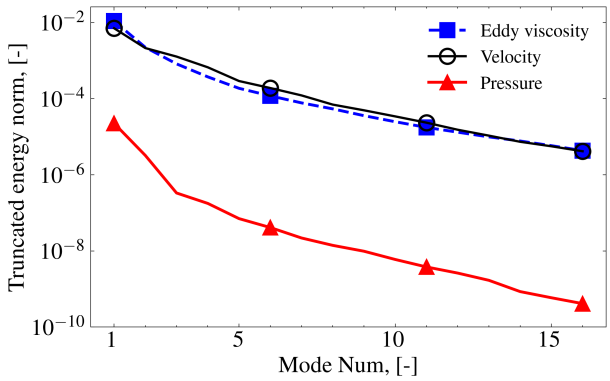


Figure 4.30: POD relative truncation energy ($\mathcal{E}^2$, see equation (3.7)) of velocity, pressure, and eddy viscosity against the number of modes for the wire-wrapped rod bundle. $N_{\text{train}} = 30$.

More features can be seen for the POD modes. The first six dominant velocity features are visualized in Fig. 4.31. The first to third modes capture the flow around the narrow gaps between the wires and rods, but also involve the development of the flow along the channel. The fifth mode mainly represents the flow across the narrow gap between rods and wires. The fourth and sixth modes focus on the flow near the entrance, which is related to the inflow conditions. Consequently, it can be concluded that the parameterized inlet profiles and wires significantly influence the flow patterns.

4.3.2.2 Reduced order models

The POG-Galerkin and PODI ROMs are described with the standard procedures. They are presented individually first and then compared.

POD-Galerkin ROMs Let us first discuss the POD-Galerkin ROMs that employed lifting functions and the supremizer. The influence of the size of the training set is evaluated with $N_{\text{train}} = 5, 10, 20,$ and 30 and $N_{\text{test}} = 20$. The results are illustrated in Fig. 4.32, presenting the variances of $\overline{\epsilon_{\mathbf{u}}}$ against the number of modes. The sampling is plotted in D.14. The number of pressure modes is fixed to 2, and those of velocity and eddy viscosity modes vary from 2 to 10.

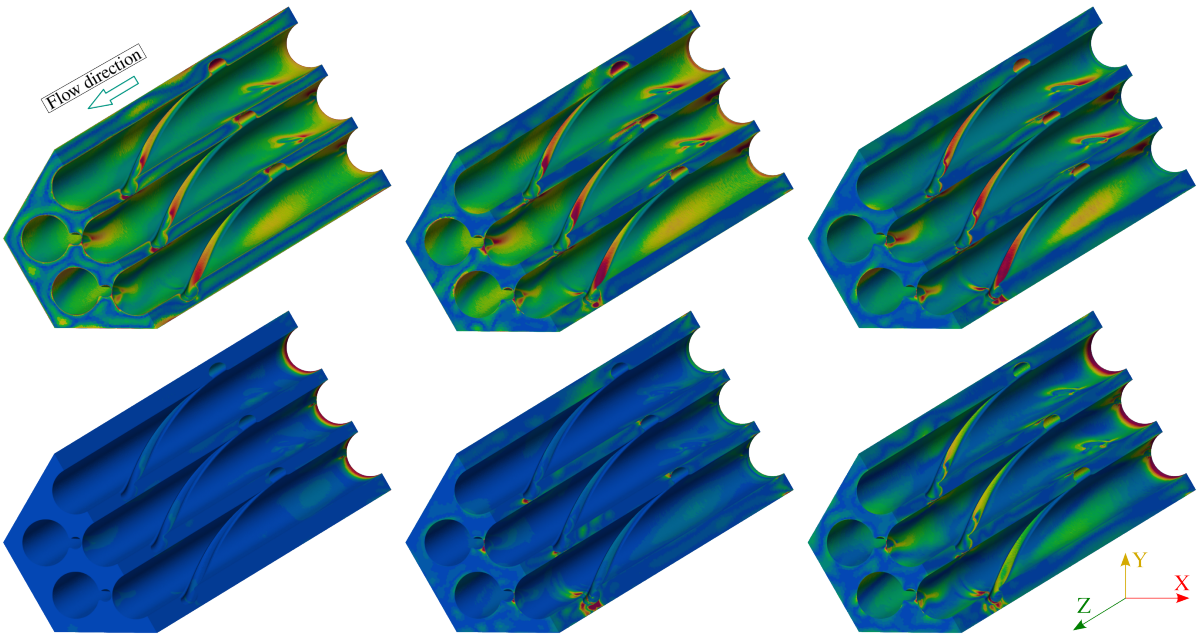


Figure 4.31: First six dominant features of velocity magnitude for the wire-wrapped rod bundle case. $N_{\text{train}} = 30$. First row from left to right: modes No. 1, 2, and 3; Second row from left to right: modes No. 4, 5, and 6.

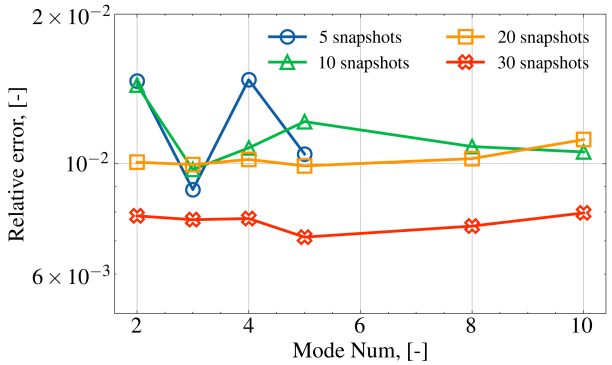


Figure 4.32: Mean relative L^2 error of velocity (POD-Galerkin based ROM utilizing lifting functions and the supremizer) with different sizes of training set, i.e., $N_{\text{train}} = 5, 10, 20, 30$. The same test samples $N_{\text{test}} = 20$ are used to quantify the ROMs. The x-axis denotes the number of $N_{\mathbf{u}}^{\text{RB}}$ and $N_{\nu_t}^{\text{RB}}$, and the number of pressure modes, N_p^{RB}, is fixed to 2.

As expected, due to the exponential decay of POD truncation energy, POD-Galerkin ROMs with a few modes can provide acceptable approximation. When $N_{\text{train}} = 5, 10,$ and 20, the lines are close, and the value is around 10^{-2}. That is very encouraging for engineering applications. Although the improvement is not significant, ROM performance is better as N_{train} increases. When $N_{\text{train}} = 30$, the error reduces to approximately 7×10^{-3}, which is the best.

PODI ROMs Analogous results for PODI ROMs with RBF, GPR, and ANN are displayed hereafter. The focus is still on the influence of the size of the training set. Fig. 4.33 shows the results of POD-RBF and POD-GPR ROMs with $N_{\text{train}} = 5, 10, 20,$ and 30 and $N_{\text{test}} = 20$.

A comparison of the different kernels is presented in Appendix D (see Figs. D.15 and D.16). Based on the analyses, the best kernels are employed to build surrogates, i.e., *Thin_plate_spline* or *Quintic* for RBF and *RBF* or *Matern* for GPR.

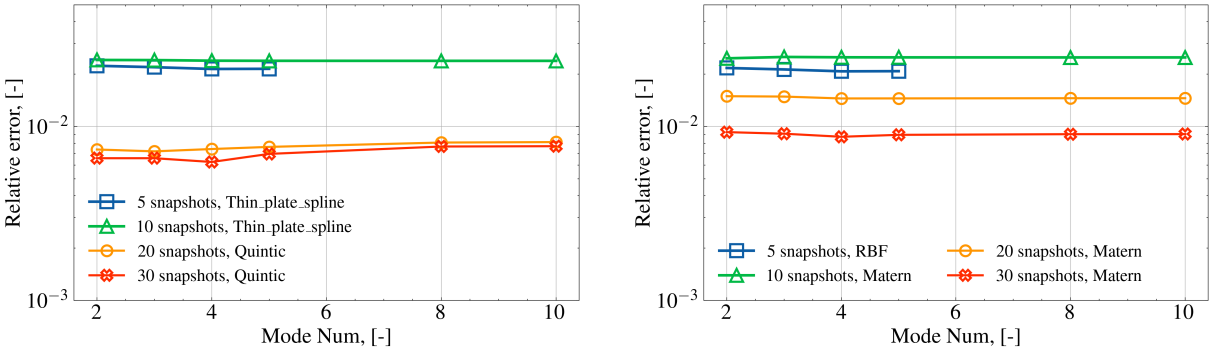


Figure 4.33: Mean relative L^2 error of velocity of PODI ROMs with $N_{\text{train}} = 5, 10, 20,$ and 30 against the number of $\mathbf{u}$ and ν_t modes. Left: RBF with *Thin_plate_spline* or *Quintic* kernel; Right: GPR with *RBF* or *Matern* kernel. Quantified by the common test set $N_{\text{test}} = 20$.

The general trend observed in Fig. 4.33 is that the accuracy is improved as N_{train} increases. An interesting point is that the results of $N_{\text{train}} = 5$ are better than those for $N_{\text{train}} = 10$. After checking the case-specific errors, it is found that, when $N_{\text{train}} = 10$, the predictions are much worse for the cases with low inflow conditions, i.e., μ_1^L or $\mu_2^L \approx 0.6$, whose errors are around 0.03. Thus, the average is affected.

Additionally, the lines in Fig. 4.33 are almost unchanged when the number of modes increases. For PODI techniques, the error comprises two parts: the POD truncation error and the interpolation error. The former is reduced as the number of modes increases, while the latter is independent of it. The accuracy gain from more modes is counteracted by interpolation error, resulting in unchanged results.

For PODI-ANN, the influence of training set size on the results is plotted in Fig. 4.34. The architecture is fixed to 3 hidden layers and 10 neurons per layer with the *Tanh* activation function, as determined based on the analyses in Appendix D, indeed, Figs. D.17 and 4.12.

The accuracy is significantly improved as N_{train} increases. The error range is from 10^{-1} to 10^{-3}, a significant difference compared to the previous models. The lines for 20 and 30 samples are close for low-order modes. While the error increases for high-order modes when $N_{\text{train}} = 20$, indicating that the increase in surrogate error affects the overall accuracy.

4.3.2.3 Comparisons and discussions

Lastly, the performance of all ROMs varying with the size of the training set (i.e., $N_{\text{train}} = 5, 10, 20,$ and 30) is compared in Fig. 4.35. The accuracy of velocity fields is validated by high-fidelity solutions in the test set ($N_{\text{test}} = 20$). The projection error $\mathcal{E}$ is included for reference.

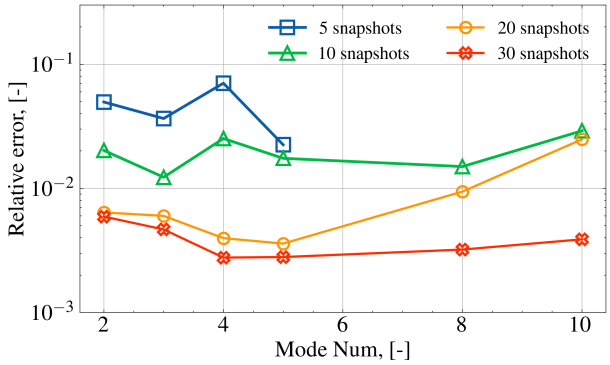
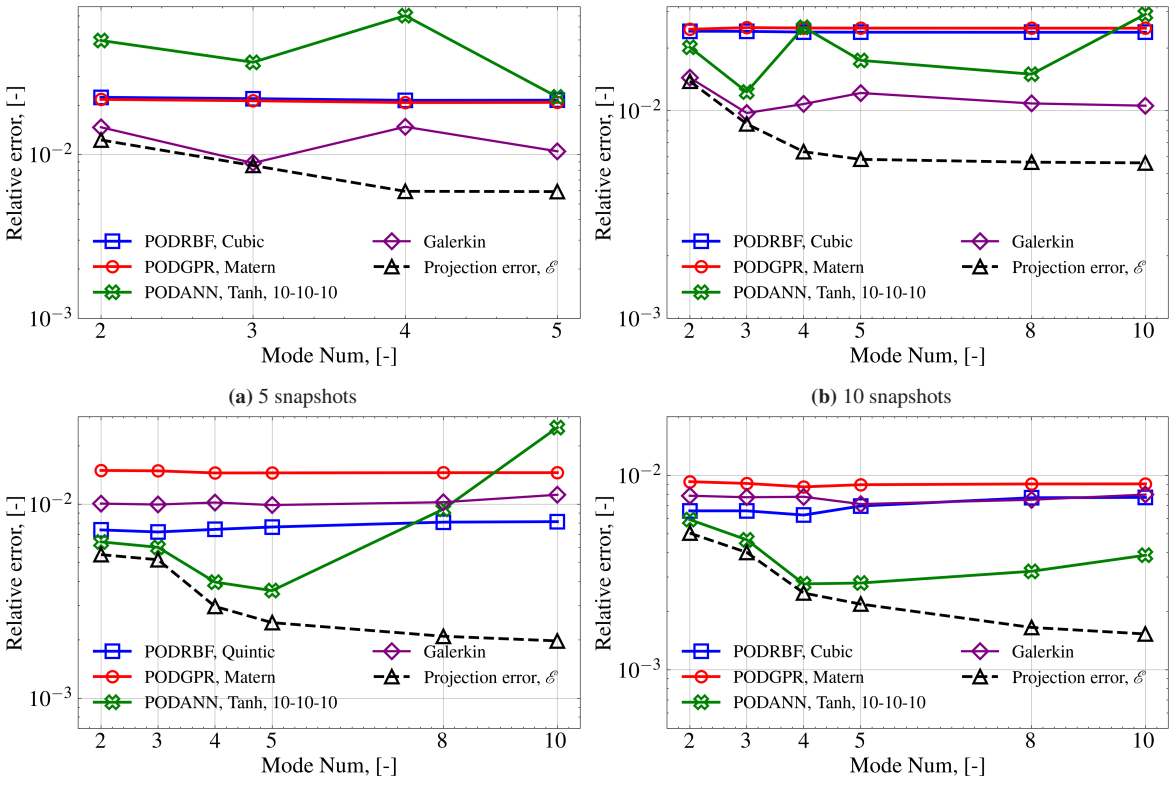


Figure 4.34: Mean relative L^2 error norm of velocity of PODI ROMs with $N_{\text{train}} = 5, 10, 20,$ and 30 against the number of $\mathbf{u}$ and ν_t modes. ANN with *Tanh* activation function and $10 - 10 - 10$ hidden layers. Test set is $N_{\text{test}} = 20$.

The same trend can be observed as in the previous cases. POD-Galerkin ROMs are more accurate than others when $N_{\text{train}} = 5$ and 10, indicated in Figs. 4.35a and 4.35b. When employing 20 samples, ANN provides the best predictions for low-order modes, but the error increases when $N_{\mathbf{u}}^{\text{RB}} > 5$, as shown in Fig. 4.35c. For $N_{\text{train}} = 30$, the line of ANN is the best and very close to the projection error (see Fig. 4.35d), which emphasizes again that PODI-ANN is a promising method when sufficient FOM data is available.



(a) 5 snapshots

(b) 10 snapshots

(c) 20 snapshots

(d) 30 snapshots

Figure 4.35: Comparisons of POD-Galerkin and PODI ROMs for the wire-wrapped rod bundle case with different sizes of training set. The projection error is also included for reference. The results are obtained with $N_{\text{train}} = 5, 10, 20, 30$, and validated with a standard test set $N_{\text{test}} = 20$.

The computational cost is summarized in Table D.3. The speedup ratio is in the order of 10^6 when considering only the solving time. It is in the range of 10^2 to 10^3 when including POD and ROM construction (i.e., projection for POD-Galerkin and training for PODI).

4.4 Conclusions

In this chapter, the POD-Galerkin and PODI ROMs are used to analyze three cases of increasing complexity: a 2D channel, a 3D seven-pin bare rod bundle, and a 3D seven-pin wire-wrapped rod bundle. The viscosity and inlet velocity profiles are parameterized to generate different flow conditions.

The focus is on the influence of the training set size, i.e., $N_{\text{train}} = 5, 10, 20,$ and 30. The same test set ($N_{\text{test}} = 20$) is used to validate the ROMs. Besides, the kernels for RBF and GPR, as well as the ANN architecture, are also evaluated. The main findings are summarized as follows. Note that some detailed results are presented in Appendix D for conciseness and clarity.

Several common aspects are worth summarizing.

- In all three cases, the POD truncation energy decays exponentially, demonstrating the existence of dominant flow features. That is very encouraging for engineering applications.
- For POD-Galerkin ROMs, the influence of boundary treatments and stabilization methods is not noticeable. All four formulations can result in similar accuracy. For PODI ROMs, the choice of kernels (RBF and GPR) and the ANN architecture can significantly affect the results. These configurations should be carefully determined by numerical experiments.
- The size of the training set can affect the results of ROMs. The performance is improved as N_{train} increases. POD-Galerkin ROMs are more robust when the data is insufficient. PODI ROMs are enhanced with large datasets. ANNs can achieve the best predictions when more data is provided.
- The number of modes is an important factor for both types of ROMs. The error of a ROM consists of two parts, i.e., the truncation of POD and the error resulting from the ODE solver or interpolation. The former is reduced as the number of basis functions increases. However, the latter might increase or be independent of it, leading to different trends of the overall quantities. Generally speaking, the value decreases as the number of modes increases. Then, it reaches a certain level and remains nearly unchanged. However, if the reduced basis dimension is large, the ROM approximations of dominant features may become less accurate, and offset the accuracy gain from more modes.

5 Local ROM for fluid dynamic analysis of a typical rod bundle

This chapter illustrates the application of the PODI-based local ROM to the fluid dynamic analysis of a seven-pin wire-wrapped bundle.

The implementation contains three parts. The first step concerns domain decomposition to identify the generic geometrical pattern in the bundle. Then, the strategy for extracting subdomain-level snapshots and computing the local POD modes is presented. Once the reduced basis is available, the last procedure is to construct the local ROM. Similar to the global approach, the interpolants can be built using RBF, GPR, and ANN. Since the configurations (e.g., kernel, network architecture) of those regression models are varied, sensitivity analyses are conducted to evaluate their influence. Two cases are adopted for the ROM evaluation: (i) the geometry generating snapshots; (ii) a model larger than the training one. The second extended domain is used to demonstrate the capability and generality of the local ROM in assembling a larger domain with repeating partitions.

Be aware that decomposition for a subchannel and bare rod bundle is straightforward, and thus, this chapter is only for the most challenging geometry, the wire-wrapped rod bundle.

5.1 High-fidelity simulations

The 7-pin wire-wrapped rod bundle has already been employed for the investigation of the global ROM in Section 4.3. Thus, only a brief description is provided for completeness.

The geometry and boundary conditions are shown in Fig. 4.26. In general, the velocity inlet and pressure outlet are imposed, and all walls are treated as no-slip boundaries. Two sets of parameters are considered, namely, viscosity and inflow profiles.

The mesh with approximately 36 million elements is used to capture the complex flow features around the wires and rods (see Fig. 4.27).

Several high-fidelity results are visualized in Fig. 4.28 and 4.29, including velocity magnitude and streamlines.

Those already mentioned in Section 4.3 are omitted. The following sections illustrate only the procedures for domain decomposition, snapshot collection, local POD basis generation, and local ROM construction.

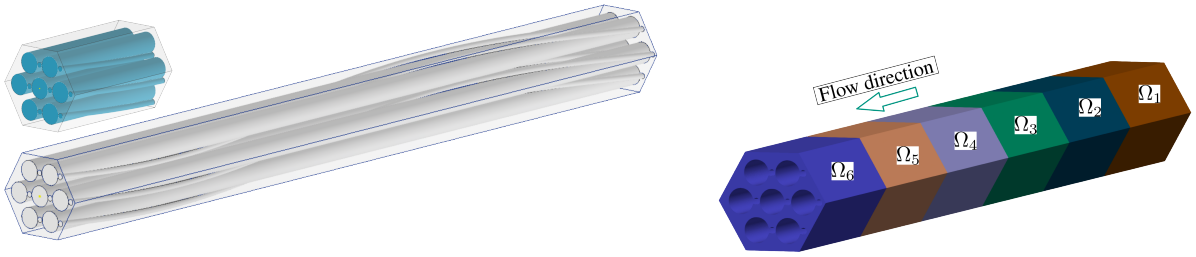
5.2 POD and dominant features

This section will first clarify how the wire-wrapped bundle is decomposed into several subdomains. The generic partition is identified. Since the global solutions have already been computed, the local fields within each subdomain are mapped to the reference domain and collected as snapshots. Dimensionality reduction is achieved by POD.

5.2.1 Domain decomposition

The general concept to decompose a rod bundle has been illustrated in Fig. 1.8. Here, the focus is on the division of an industrial relevant computational domain – a 7-pin wire-wrapped bundle with a longitudinal length of one wire-pitch –.

The geometry of the bundle and its division is depicted in Fig. 5.1. The total length of the bundle equals one wire pitch. The outer surface of the channel is a hexagonal prism. Rods are arranged in a hexagonal pattern within the channel. Wires are wrapped around each rod in a helical pattern. Consequently, considering the geometrical periodicity and symmetry, it can be decomposed into six blocks, which are noted $\Omega_i,\ i = 1, \ldots, 6$ from entrance to exit. Each block is one-sixth of the bundle.



(a) (b)

Figure 5.1: Domain decomposition of a 7-pin wire-wrapped bundle. (a) One-sixth of the bundle and the global domain. (b) Six decomposed subdomains within the bundle. They are labeled $\Omega_i,\ i = 1, \ldots, 6$ from the entrance to exit.

5.2.2 Snapshots and local POD modes

There are two strategies for collecting snapshots: individual and generic. Consequently, different POD modes are computed and then compared.

5.2.2.1 Snapshots collection

The first strategy is to gather the local solutions from each subdomain separately. Thus, six snapshot matrices will be generated. The procedure can be expressed as follows. Suppose there are N_μ

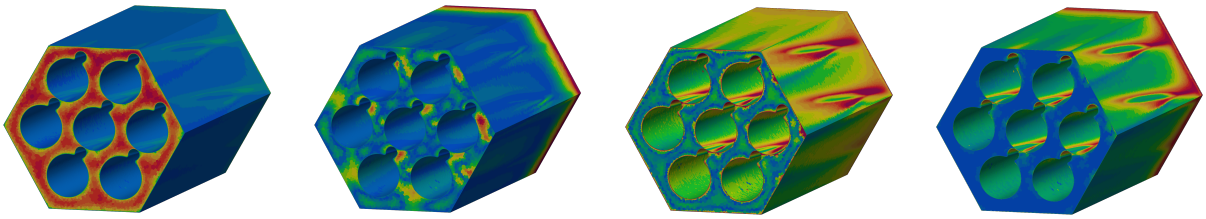
parameter instances, then for the i-th subdomain (Ω_i), the velocity fields are denoted as $\mathbf{u}_i(\mu_j)$, $j = 1, 2, \ldots, N_\mu$. The corresponding snapshot matrix is $\mathbf{S}_i = \left[\mathbf{u}_i(\mu_1), \mathbf{u}_i(\mu_2), \ldots, \mathbf{u}_i(\mu_{N_\mu})\right]$. Then, POD is applied to each $\mathbf{S}_i$ independently. Since six subdomains exist, six sets of basis functions will be computed. Those are called *individual modes*.

The second approach is to combine all local solutions from different subdomains into one single snapshot matrix. More specifically, for the i-th subdomain (Ω_i), the snapshot matrix is defined as above, $\mathbf{S}_i = \left[\mathbf{u}_i(\mu_1), \mathbf{u}_i(\mu_2), \ldots, \mathbf{u}_i(\mu_{N_\mu})\right]$. Then, they can be concatenated together to form a large snapshot matrix, i.e., $\mathbf{S} = [\mathbf{S}_1, \mathbf{S}_2, \ldots, \mathbf{S}_6]$. The POD is performed on $\mathbf{S}$ and obtains the so-called *generic modes*.

5.2.2.2 Individual modes

For the individual strategy, six sets of basis functions are generated. To avoid redundancy, only the modes from the first and second subdomains (Ω_1 and Ω_2) are shown and discussed here.

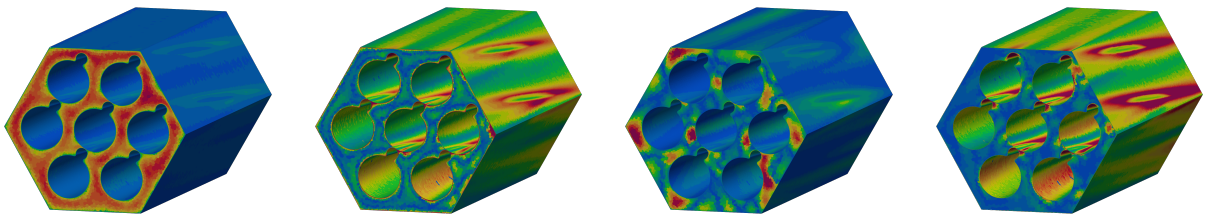
The first four individual POD modes of Ω_1 are displayed in Fig. 5.2. The average flow pattern characterizes the first fields. The second and fourth capture the inlet effects. The third mode represents the swirling motion induced by the wires.



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

Figure 5.2: First four individual POD modes from Ω_1. (a) Mode 1. (b) Mode 2. (c) Mode 3. (d) Mode 4.

The first four dominant features of Ω_2 are illustrated in Fig. 5.3. The first also represents the mean flow, and its distribution is similar to that in Ω_1, with the peak values in different locations than in Fig. 5.2a. The other three modes denote the flow structures induced by the wires. The inlet effects do not explicitly appear in the modes.



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

Figure 5.3: First four individual POD modes from Ω_2. (a) Mode 1. (b) Mode 2. (c) Mode 3. (d) Mode 4.

Similarities can be found when the two sets of flow patterns are compared. The first dominant mode in both subdomains represents the average flow, and it is similar in Ω_1 and Ω_2. The presence of wire introduces swirling motion, which is captured in both cases. However, the detailed structures and locations of the peak values are different. Inlet effects are specifically observed in the first block.

5.2.2.3 Generic modes

The generic POD modes computed from all blocks are shown in Fig. 5.4. They are similar but not identical to the individual ones. The first still characterizes the mean flow pattern. The second and third modes represent the existence of wires that force the bulk flow to the opposite sides. When observing higher-order modes (e.g., modes 7, 8, and 9), the inlet effects and local behavior around the wires are captured.

It may be concluded that the generic modes can extract the common features. However, patterns specific to certain partitions may be lost.

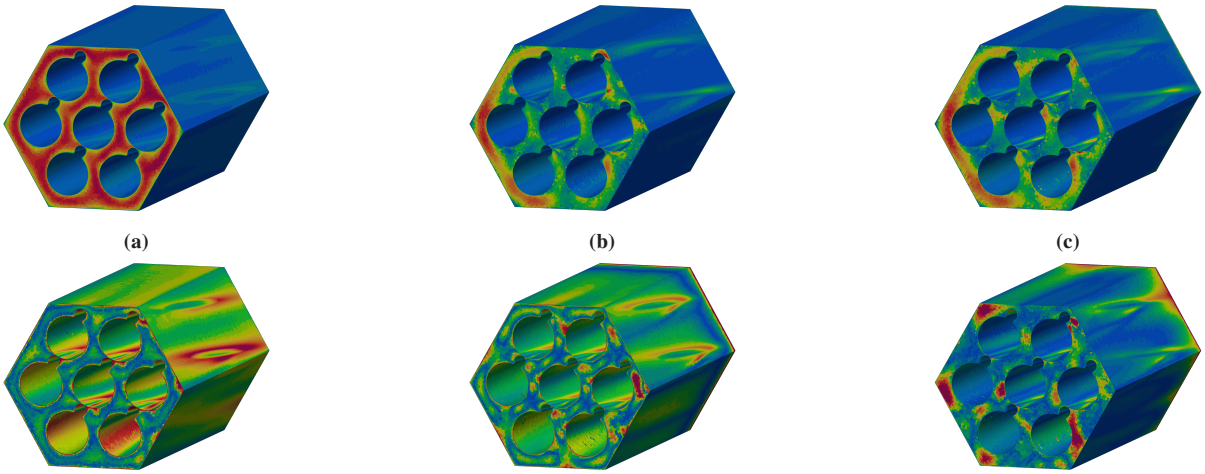


Figure 5.4: Generic POD modes from all subdomains. (a) Mode 1. (b) Mode 2. (c) Mode 3. (d) Mode 7. (e) Mode 8. (f) Mode 9.

The curves of POD truncation energy (see definition in equation (3.7)) for both individual and generic modes are plotted in Fig. 5.5. The two lines for Ω_1 and Ω_2 are similar. They both decrease exponentially as the number of modes increases. The generic strategy has a higher initial value and exhibits a slower decay rate. The line for generic modes is two or three orders of magnitude higher than the line for the individual. The results are reasonable, since the generic modes need to capture common features across all subdomains, which is more challenging than the individual ones.

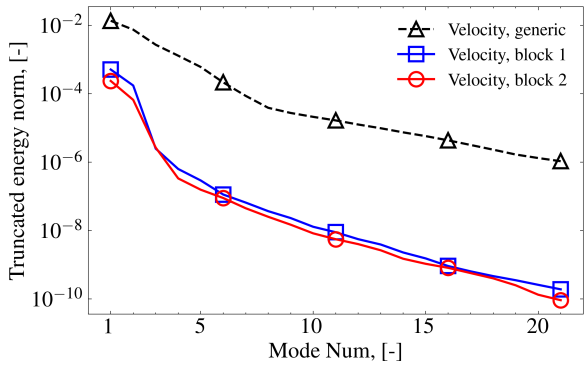


Figure 5.5: POD relative truncation energy ($\mathcal{E}^2$, see equation (3.7)) of velocity against the number of modes for the decomposed wire-wrapped rod bundle. The three lines represent individual modes from Ω_1 and Ω_2, and the generic modes from all subdomains. $N_{\text{train}} = 30$.

Note that all the above results are obtained using 30 training samples. The effects of training set size on the POD modes are not shown for brevity. In general, the dominant features are not significantly influenced by the number of snapshots, while the higher-order modes may vary.

5.3 Local ROM construction and sensitivity analyses

5.3.1 Preliminaries

In this research, the aim is to demonstrate the possibility of constructing local ROMs for a geometry consisting of repetitive blocks. Thus, the generic modes are employed for the following ROM modeling. The individual modes can be adopted in a similar but more straightforward manner.

The local ROMs are built using the PODI framework because it is non-intrusive and convenient for practical applications. In addition, a comparison of global ROMs have already shown that PODI achieves satisfactory accuracy. Three regression models, namely RBF, GPR, and ANN, are utilized to construct the interpolants for the modal coefficients.

Sensitivity analyses are conducted to evaluate the effects of training set size and regression model configurations. The sizes of the training set, N_{train}, are varied to 5, 10, 20, and 30. The number of test samples, N_{test}, is fixed at 20, the same as in the global ROM case. For RBF and GPR, several commonly used kernel functions are tested. For ANN, different architectures (i.e., the number of hidden layers and the number of neurons per layer) are evaluated.

5.3.2 Radial basis function

Now, let us first investigate the performance of RBF-based local ROMs. Four kernels, i.e., *Thin_plate_spline*, *Cubic*, *Multiquadric*, and *Gaussian*, are compared in Fig. 5.6. The projection error is presented for reference. When adopting 15 modes, the projection error is still around

1%, which is not as good as the global strategy that achieves 2×10^{-3}. However, the accuracy is good enough for many engineering applications. Additionally, this is acceptable, as the focus here is on constructing local ROMs.

It can be observed that the errors of those kernels behave distinctly. The lines of the thin-plate spline and cubic kernels decrease for the POD modes less than 8, and then they start to grow. The overall trends for the multiquadric and Gaussian kernels are continuously decreasing. It may be concluded that the multiquadric kernel outperforms the others. When $N_u^{\mathrm{RB}} = 15$, the projection error is around 1%, and the multiquadric kernel can reach 2%, which demonstrates the feasibility of RBF interpolation for the coupling algorithm.

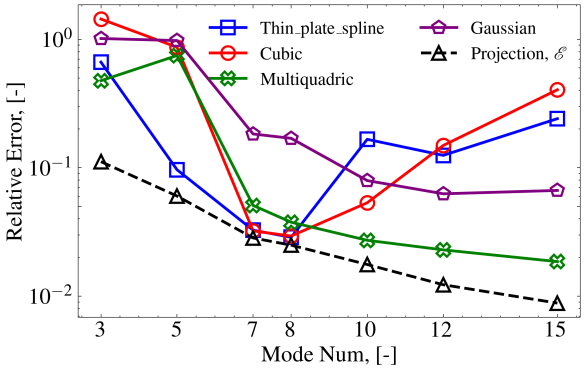


Figure 5.6: Mean relative L^2 error of the ROM velocity against the number of modes. Four RBF kernels (i.e., *Thin_plate_spline*, *Cubic*, *Multiquadric*, and *Gaussian*) are compared with $N_{\mathrm{train}} = 30$. $N_{\mathrm{test}} = 20$. The projection error is shown as a reference.

As shown in Fig. 5.6, the training set size is fixed at 30. To further explore its effect, Fig. 5.7 presents the results using varied N_{train}. Since *Thin_plate_spline* performs better than others when N_u^{RB} is 5, 7, and 8, it is selected for comparison with *Multiquadric*.

Fig. 5.7 indicates that when only five training samples are used, *Multiquadric* decreases slowly and remains at a high level. Its accuracy is much worse than *Thin_plate_spline*. The curve of *Thin_plate_spline* grows exponentially after $N_u^{\mathrm{RB}} = 8$. Similar trends are observed for the other training sizes (i.e., 10, 20, and 30). But for $N_{\mathrm{train}} = 20$ and 30, *Multiquadric* is better because the error reduces continuously, while the *Thin_plate_spline* results in bad accuracy for more than 10 POD modes.

It may be concluded that *Multiquadric* is more robust than *Thin_plate_spline* when N_u^{RB} varied. Also, its quantities are close to the projection error. The best accuracy is achieved by *Thin_plate_spline* when $N_u^{\mathrm{RB}} \leq 8$, but its poor prediction for the high-order modes is unacceptable.

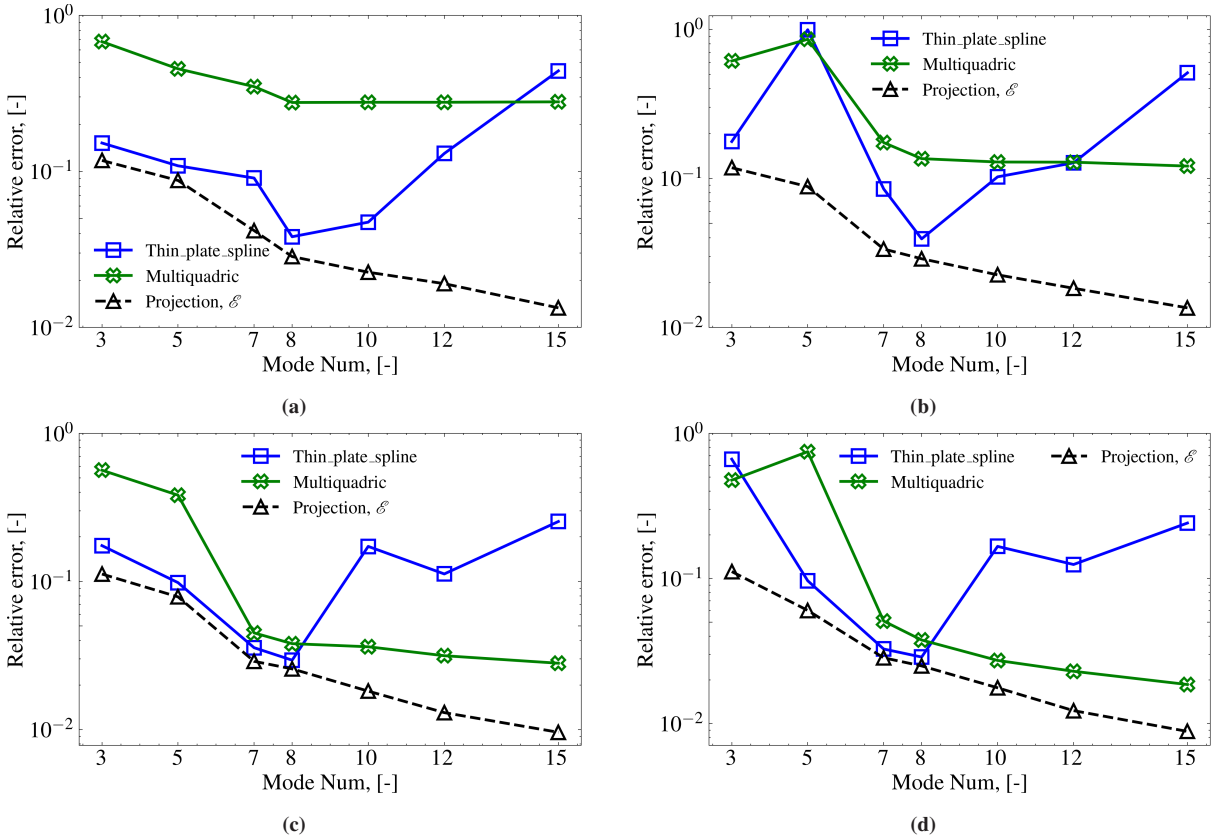


Figure 5.7: Mean relative L^2 error of the ROM velocity using PODI-RBF with varied training set sizes. Two kernels (i.e., *Thin_plate_spline* and *Multiquadric*) are compared. (a) 5 training samples. (b) 10 training samples. (c) 20 training samples. (d) 30 training samples. $N_{\text{test}} = 20$

5.3.3 Gaussian process regression

Regarding GPR-based local ROMs, two commonly used kernels, i.e., *Matern* and *RBF*, are chosen for comparison. Recall that the RBF kernel is a type of covariance function in GPR, which differs from the name of RBF interpolation. In GPR, the length scale is a crucial hyperparameter that scales the distance between two points in the parameter space. Generally, the value significantly affects the performance of the resulting surrogate model. Its effects are tested for both kernels, as shown in Fig. 5.8.

It is obvious that the *RBF* results in much higher errors than the *Matern*. Hence, RBF will be neglected in the following analyses.

For the *Matern* kernel, considering both robustness and accuracy, the length scale of 10.0 is the best choice. The line declines as the number of POD modes increases, and they are comparable to the projection error.

After selecting the *Matern* kernel with a length scale of 10.0, the variation versus errors against different N_{train} are investigated in Fig. 5.9. It can be seen that all lines decrease as the number of POD modes increases. The accuracy grows significantly as the training set size increases from

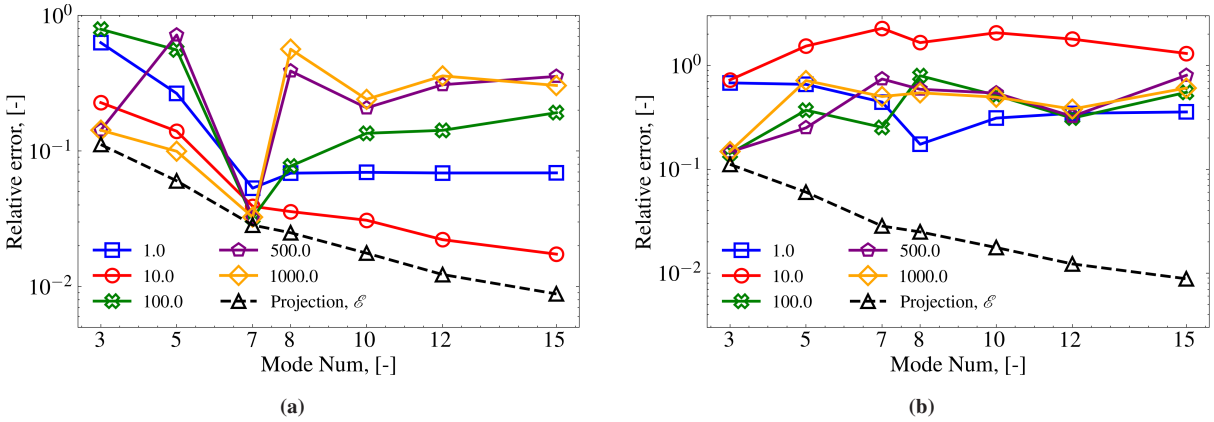


Figure 5.8: Mean relative L^2 error of velocity employing GPR-based ROMs with varied length scales, i.e., 1.0, 10.0, 100.0, 500.0, and 1000.0. (a) Matern kernel. (b) RBF kernel. $N_{\text{train}} = 30$ and $N_{\text{test}} = 20$.

5 to 20. While the results for $N_{\text{train}} = 20$ and 30 almost overlap. It demonstrates that 20 training samples are sufficient to construct a reliable GPR-based local ROM in this case.

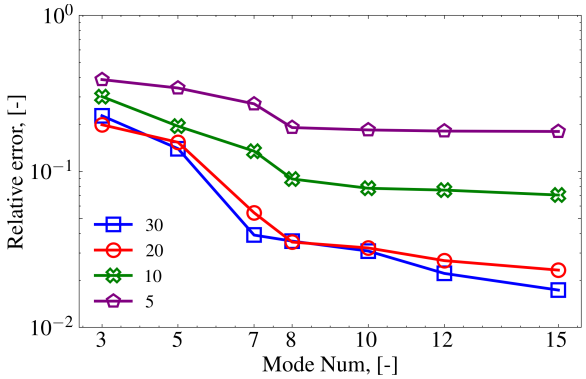


Figure 5.9: Mean relative L^2 error of velocity using the PODI-GPR ROM with different training set sizes, i.e., $N_{\text{train}} = 5$, 10, 20, and 30. The Matern kernel with a length scale of 10.0 is applied. $N_{\text{test}} = 20$

5.3.4 Artificial neural networks

The last regression model is ANN. The evaluation is to identify the influences of activation functions, network architectures, and training set sizes.

The comparison of three commonly adopted activation functions, i.e., *ReLU*, *Sigmoid*, and *Tanh*, are presented in Fig. 5.10. The network has four layers, and each contains 50 neurons. The three lines are significantly different. *Tanh* results are insensitive to the number of POD modes, and the values remain at a high level. *Sigmoid* performs better than *Tanh*, but its performance is not robust when N^{RB} varied. The best performance is accomplished by *ReLU*, which is robust and precise.

The influences of network structures are investigated in Fig. 5.11. It is well known that deeper, wider networks have better fitting capabilities for the training data. However, they may also lead to overfitting, which deteriorates generalization performance on test datasets.

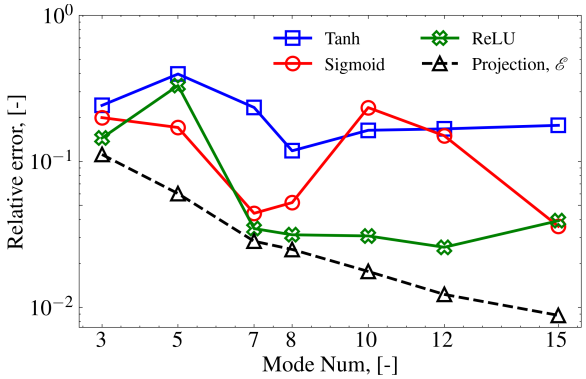


Figure 5.10: Mean relative L^2 error of the ROM velocity using the PODI-ANN. Three different activation functions (i.e., ReLU, Sigmoid, and Tanh) are compared for local ROMs with $N_{\text{train}} = 30$ and $N_{\text{test}} = 20$. The network has four layers, each with 50 neurons.

Both the number of hidden layers and the number of neurons per layer are varied to explore their effects. Since the best activation function is *ReLU*, it is fixed.

Fig. 5.11a illustrates the results using 2 to 6 hidden layers, while each layer contains 50 neurons. It can be observed that the lines for the 3- and 4-layer models are comparable and achieve the best accuracy. For 2 and 5 layers, they can reach low errors with $N_u^{\text{RB}} = 7$ or 8, but the overall trends are unstable. The worst performance is observed with 6 layers, which may be due to overfitting.

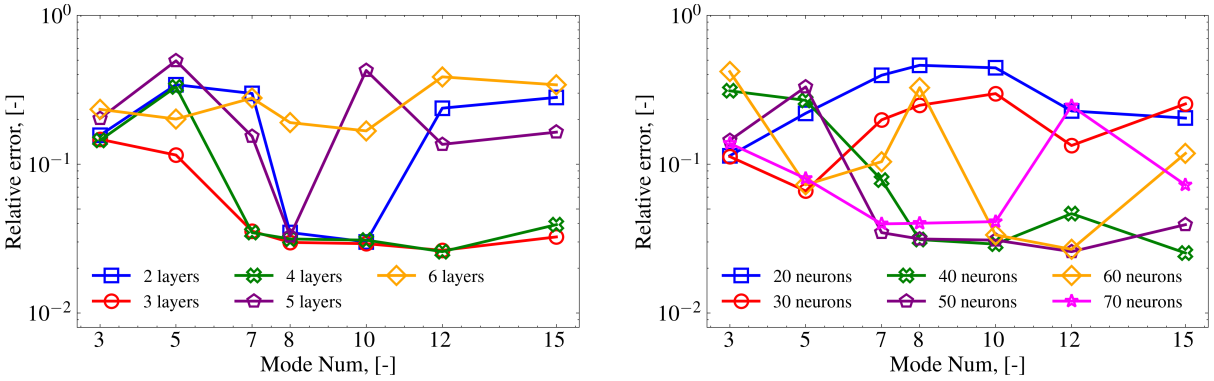


Figure 5.11: Mean relative L^2 error of the ROM velocity using different ANN architectures. (a) Number of hidden layers, i.e., 2 to 6 (50 are applied for each layer). (b) Number of neurons per layer (four layers are used), i.e., 20 to 70. $N_{\text{train}} = 30$ and $N_{\text{test}} = 20$.

Fig. 5.11b presents the results using different neurons per layer, i.e., 20, 30, 40, 50, 60, and 70. Four hidden layers are employed, which has been demonstrated in Fig. 5.11a.

There are two observations. First, when the number of neurons is small (e.g., 20 or 30), the errors are relatively high. Second, when too many neurons are used (e.g., 60 and 70), accuracy is unstable as N_u^{RB} varies. The best performance is achieved by 40 or 50 neurons per layer.

Based on the above sensitivity analyses, the ANN with the *ReLU* activation function, four hidden layers, and 50 neurons per layer (i.e., 50-50-50-50) can be regarded as the optimal configuration for local ROM construction.

Finally, the effects of training set size are assessed in Fig. 5.12. As expected, the errors decrease as N_{train} increases. 20 training samples are sufficient to obtain a local ROM with a relative L^2 error of around 3%.

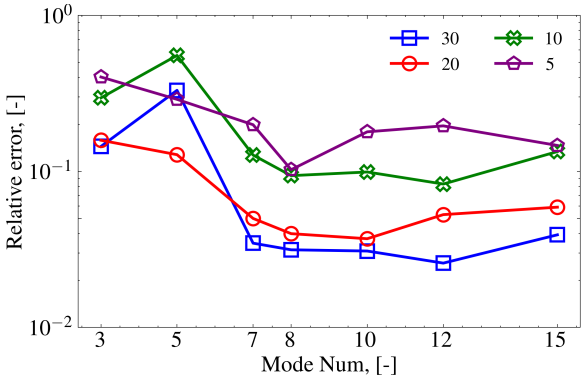


Figure 5.12: Mean relative L^2 error of the PODI-ANN ROM velocity using $N_{\text{train}} = 5, 10, 20$, and 30. The network employs the *ReLU* activation function and four hidden layers with 50 neurons per layer, i.e., 50-50-50-50. $N_{\text{test}} = 20$

5.3.5 Comparison and validation

Following the sensitivity analyses of three regression models, it can be inferred that the configurations have a significant impact on the performance of local ROMs. Now, let us compare the best results of RBF, GPR, and ANN in Fig. 5.13.

The configurations of those are listed as follows: (i) RBF: *Multiquadric* kernel; (ii) GPR: *Matern* kernel with a length scale of 10.0; (iii) ANN: *ReLU* activation function, four hidden layers, and 50 neurons per layer (i.e., 50-50-50-50).

Unlike the global ROM case, the performances of those three local ROMs are comparable for all three training sizes. For each sub-figure in Fig. 5.13, the error declines as the number of POD modes increases.

The influences of N_{train} can be summarized as follows. When $N_{\text{train}} = 5$ or 10, all three lines are relatively high. As the training set size increases to 20 or 30, the errors decrease significantly, which almost achieves the best possible accuracy. For instance, when $N_{\text{train}} = 30$ and $N_u^{\text{RB}} = 15$, all three ROMs' results are around 2%, which is very close to the theoretical limit (i.e., projection error of 1%).

It can be concluded that PODI models require sufficient training samples to achieve satisfactory interpolation precision. For example, POD modes with small training sets can already sufficiently represent the solution space (i.e., projection error is approximately 1%). However, the ROMs

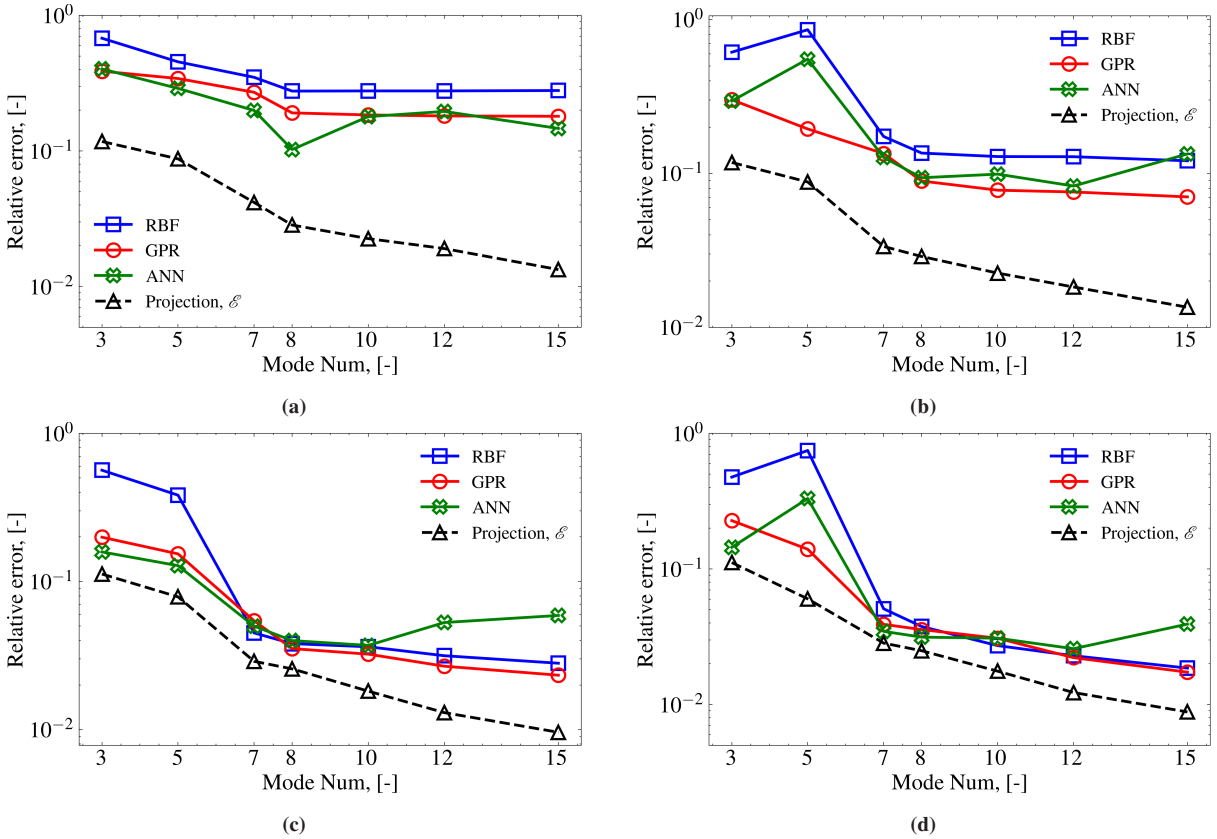


Figure 5.13: Mean relative L^2 error of the ROM velocity using three regression models with varied training set sizes. (a) 5 training samples. (b) 10 training samples. (c) 20 training samples. (d) 30 training samples. $N_{\text{test}} = 20$

cannot lead to high accuracy due to poor generalization of regression models. Additionally, all three approaches perform well and have similar capabilities for constructing ROMs in this case.

5.4 Application for extended bundle

To further discover the potential of PODI-based local ROMs, an extended 7-pin wire-wrapped bundle with a length of two wire pitches is considered. The geometry and boundary conditions are the same as the previous one, except for the longitudinal length. The domain decomposition strategy is also the same. The bundle is divided into 12 identical subdomains, as shown in Fig. 5.14.

A mesh with approximately 73 million elements is utilized for high-fidelity simulations. Due to the increased computational cost, only five cases are simulated for verification. These cases cost much more CPU time than the previous bundle. Each RANS simulation takes around 70 hours on the same platform, i.e., a single node with an Intel Xeon Platinum 8358 CPU (64 cores, 2.6 GHz).

The parameters are selected from the test set of the previous bundle. The values are listed in Table 5.1. As before, the two inflow factors and kinematic viscosity are parameterized.

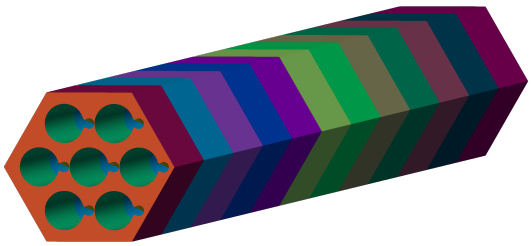


Figure 5.14: Domain decomposition of an extended 7-pin wire-wrapped bundle with a length of two wire-pitches. The bundle is decomposed into 12 identical subdomains. They are highlighted using different colors.

Table 5.1: Parameter instances for the extended 7-pin wire-wrapped bundle simulations.

Case	Inflow factor 1	Inflow factor 2	kinematic viscosity (m^2/s)
1	0.0	0.6	1.486×10^{-7}
2	1.35	0.0	1.907×10^{-7}
3	0.675	0.675	2.327×10^{-7}
4	0.155	0.753	1.583×10^{-7}
5	0.503	0.304	1.898×10^{-7}

Since the procedures for snapshot collection, POD-based generation, and local ROM construction have already been illustrated, only the final results are presented here. The best configurations of the three regression models identified before are adopted directly. The comparisons of three local ROMs with varied POD modes are shown in Fig. 5.15. The training set size is fixed at 30, and the test set contains five cases.

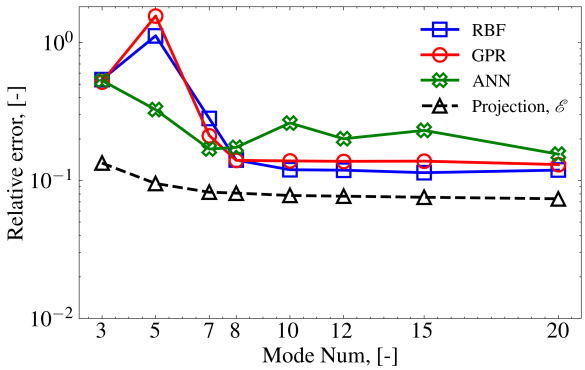


Figure 5.15: Mean relative L^2 error of the ROM velocity for the 7-pin wire-wrapped bundle with 12 subdomains using three regression models (i.e., RBF, GPR, and ANN). The training set size is 30, and the test set contains five cases (see Table 5.1).

Generally, the errors decrease as the number of POD modes increases. But when N_u^{RB} is larger than 8, the projection error is around 8%, and the value almost remains constant when N_u^{RB} exceeds 8. Similar trends are observed for all three regression models. The accuracy of the three local ROMs is comparable, and their values are around 13% when $N_u^{\mathrm{RB}} = 8$. The best performance is reached with RBF, with an error of approximately 12%.

Although the accuracy of the extended bundle is not as good as the previous one, it is still acceptable for many engineering applications. More importantly, the results are not far from the theoretical limit (i.e., projection error). It demonstrates the perspective of utilizing the local framework for practical problems involving complex geometries with repetitive patterns.

Despite the errors, the results indicate that the iterative local ROMs can be conveniently applied to extended domains without additional high-fidelity simulations. It has the potential to significantly reduce the computational cost for the offline stage of ROM construction.

To improve the accuracy of the local ROM, there are many possible solutions. For instance, employing better strategies for snapshot collection, increasing the number of training samples, and adopting advanced regression models. In particular, the projection error is relatively high in this case, which limits the best achievable accuracy of local ROMs. Consequently, enhancing the representation capability of the POD basis is crucial. These aspects are worth investigating in future works.

5.5 Summary and conclusions

In this chapter, the application of PODI-based local ROMs to the fluid dynamic analysis of a typical rod bundle has been illustrated. The 7-pin wire-wrapped rod bundle with one wire pitch length is decomposed into six identical subdomains. The generic strategy is adopted to collect snapshots and compute the POD basis. The local reduced systems are constructed using three algorithms, i.e., RBF, GPR, and ANN. Sensitivity analysis is performed to evaluate the effects of configuration for each model. Comparisons among those three models are also conducted with varying training set sizes.

The best settings for those three algorithms have been identified. They are

- Radial Basis Function (RBF): *Multiquadric* kernel;
- Gaussian Process Regression (GPR): *Matern* kernel with a length scale of 10.0;
- Artificial Neural Network (ANN): *ReLU* activation function, four hidden layers and 50 neurons per layer (i.e., 50-50-50-50).

There are several conclusions:

- The Dirichlet-Neumann coupling strategy is effective for constructing local ROMs for geometries with repetitive blocks. The accuracy can reach the same level as the projection error when sufficient training samples are used.
- The local ROMs can be conveniently applied to extended geometries with more repetitive blocks. Thus, the framework has a great potential to be a high-fidelity alternative to subchannel analysis for nuclear thermal-hydraulics applications.

- The performance of PODI-based ROMs significantly depends on the setup of interpolation models. When optimal configurations are selected, all three frameworks achieve satisfactory accuracy.
- The local generic POD modes can extract common features from all subdomains. However, the detailed patterns specific to certain partitions may not be well represented

6 Summary

6.1 Overview

In this dissertation, the development and application of non-intrusive Reduced Order Models (ROMs) for fluid dynamic analyses of typical rod bundles in liquid metal-cooled fast reactors have been presented.

The narrative begins with a background introduction in Chapter 1. The fundamentals of liquid metal-cooled fast reactors and the significance of thermal-hydraulic analysis are discussed. The implementation of numerical techniques for nuclear reactors is introduced. To address the high computational cost of CFD for many-query problems, the ROM concept is proposed. The basic ideology and classifications of ROMs are introduced. The local ROMs based on domain decomposition methods are highlighted to address challenges posed by large-scale geometries. Considering the motivation for fast parametric computation in nuclear engineering, the objectives and structure of this dissertation are also outlined.

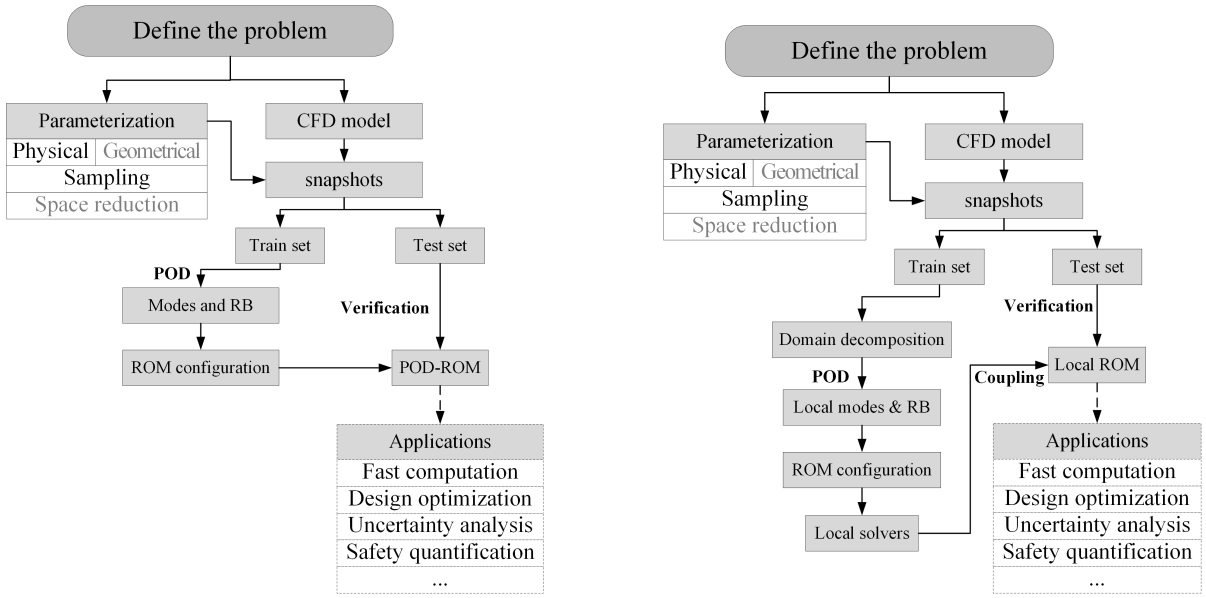
The state-of-the-art review of the topic corresponding to this research is provided in Chapter 2. The existing thermal-hydraulic tools for nuclear reactors are reviewed, including the system code, subchannel approach, and CFD method. The current status of CFD applications for the reactor core simulation is summarized. Then, the procedures and categorization of ROMs are explained, including classical POD-Galerkin ROMs and emerging non-intrusive data-driven ROMs. References regarding the ROM applications for nuclear reactor analyses are specifically reviewed. To identify the local ROM strategy for the current study, a comprehensive literature review of local ROM (i.e., the combination of domain decomposition and ROM) is conducted. The existing methods are categorized into several groups. Considering implementation feasibility and the characteristics of the target problems, iterative frameworks are selected for developing local ROMs in this dissertation.

Chapter 3 presents the methodologies and theories employed in this project. The Reynolds-averaged Navier-Stokes equations, the finite volume method, and the CFD solver based on OpenFOAM are introduced. The preliminary knowledge for constructing ROMs is described in detail, including the snapshot generation and basis computation via proper orthogonal decomposition. The mathematical formulations of POD-Galerkin ROMs and several machine-learning algorithms for POD with interpolation ROMs are explained. The strategy for coupling local ROMs, the Dirichlet-Neumann iterative method, is then illustrated.

The application of the global ROM is presented in Chapter 4. The procedures are summarized in Fig. 6.1a. Three typical geometries of rod bundles are considered as the test cases: a center subchannel, a 7-pin bare rod bundle, and a 7-pin wire-wrapped rod bundle. Sensitivity analyses for the ROM configurations are conducted. The results indicate that both POD-Galerkin and PODI ROMs can result in acceptable accuracy in reproducing high-fidelity solutions while significantly reducing computational cost. The advantages and limitations of each ROM technique are discussed.

In Chapter 5, the implementation of local ROM (see Fig. 6.1b) for 7-pin wire-wrapped rod bundles is presented. The domain decomposition strategy is explained, and the construction of local ROMs is described. The PODI technique is employed to approximate the solutions in each subdomain. The Dirichlet-Neumann iterative algorithm is then utilized to couple the local solvers. Various comparisons are made among different regression methods. It can be concluded that the iterative scheme is effective in ensuring the consistency between the local ROMs and achieving satisfactory accuracy.

Finally, this chapter summarizes the main findings of this research, highlights the scientific contributions, and discusses prospects.



(a) Global ROM **(b)** Local ROM

Figure 6.1: Flow charts for global and local ROMs. Only fast computations with ROMs are carried out in this research. Once ROMs have been constructed, they can be used for various applications, such as design optimization, uncertainty analysis and safety quantification. These studies are not involved in this study.

6.2 Conclusions

The major conclusions drawn from this dissertation are summarized as follows:

- For the global ROMs of typical rod bundle problems, both POD-Galerkin and PODI methods can achieve good accuracy in predicting the velocity fields. The mean relative errors of velocity are generally within 1% for the test cases considered.

  Global ROMs are applied to engineering-level problems with complex geometries, i.e., 7-pin wire-wrapped rod bundles. POD-Galerkin can achieve higher accuracy for a minimal number of snapshots. When sufficient snapshots are available, PODI can reach comparable accuracy while being easier to implement.

- The local ROM strategy based on domain decomposition and Dirichlet-Neumann iterative coupling can handle large-scale problems. It can accurately reproduce CFD solutions, with mean relative errors of around 2%. When applying the method to extended geometries with more repetitive blocks, the accuracy remains satisfactory, indicating the potential of the approach for industrial applications.

- The configurations of ROMs significantly influence the accuracy of the results. For POD-Galerkin ROMs, three aspects are crucial: boundary condition treatment, stabilization techniques, and turbulence modeling. For PODI ROMs, the selection of regression methods significantly affects the prediction performance, which should be determined by numerical tests.

6.3 Scientific contributions

There are two main scientific contributions in this dissertation:

- Both intrusive and non-intrusive ROMs are successfully developed and applied to typical rod bundle geometries, which are typical parts of nuclear reactors. The performance of both ROM techniques is systematically evaluated and compared across test cases of increasing complexity. The findings provide valuable insights into the capabilities and limitations of these ROM methods for thermal-hydraulic analyses of nuclear reactor components.

- A local ROM framework based on domain decomposition and Dirichlet-Neumann iterative coupling is proposed for large-scale models. The approach is demonstrated through a 7-pin wire-wrapped rod bundle case. The local ROM can predict high-fidelity solutions in the training geometry and also in a larger domain with more repeating subdomains. It is a promise framework for extending ROM applications to more complex, larger-scale problems in nuclear engineering.

6.4 Prospects

Several prospects for future work based on the current study are proposed as follows:

- In this study, only physical parameters (i.e., inlet velocity and viscosity) are considered as the input parameters. In future work, geometric parameters (e.g., rod diameter, pitch, and wrap wire characteristics) can be included to enhance the versatility of the ROMs.
- The scalability of the local ROM framework can be further tested for models with more subdomains and more complex geometries. It is expected that the local approach can develop to be an alternative to subchannel analysis for large-scale reactor modeling, which can provide high-resolution results while maintaining computational efficiency.
- The coupling of local ROMs can be further improved by adopting more advanced strategies, e.g., Robin-Robin coupling, optimized Schwarz methods, etc.
- Regarding the results for extended geometries using the local ROM approach, the accuracy of local ROMs can be further improved. This can be achieved by adopting advanced strategies to generate more representative local bases. Additionally, advanced regression models, such as deep neural networks, can be investigated to improve predictive performance.
- POD has been widely employed for dimensionality reduction in this dissertation. Other techniques, such as *Autoencoder* (a neural network) and dynamic mode decomposition, can be explored and compared for ROM construction.
- Fluid dynamics problems with heat transfer and/or buoyancy effects can be investigated to extend the applicability of the developed ROMs.
- The local ROM can be achieved via the POD-Galerkin approach and compared with the PODI-based local ROM presented in this dissertation.
- Recently, the adaptive framework has become popular due to its ability to maintain accuracy and reduce computation costs when generating high-fidelity solutions.

Moreover, according to the available experiences and observed references, a comprehensive process of ROMs for practical applications is proposed and shown in Fig. 6.2. Several aspects are considered and configured. The full process has not been demonstrated in this dissertation.

- CFD models are verified following well-established procedures, e.g., comparing with experiments or benchmark solutions [1].
- Domain decomposition can be integrated in two ways: either high-fidelity simulations are performed in the reference subdomains to extract local RBs, or the FOM solutions of the global domain are obtained and decomposed.
- Different strategies regarding parameterization can be adopted, e.g., physical parameters, geometric parameters, sampling strategies, parameter space reduction, etc.

[1] See [34; 35] for the instruction of V&V for CFD models.

- A ROM can be built based on an intrusive or non-intrusive approach and their configurations should be quantified via numerical tests.
- *Adaptive enrichment* is an advanced technique that balances the cost of FOM simulations with accuracy. This methodology uses an error estimator to determine the accuracy of the reduced order model (ROM) and adds new snapshots iteratively.
- *Hyper-reduction* is a technique that further reduces the computational cost of ROMs by approximating the nonlinear terms in the governing equations. This can be achieved through methods such as the *Discrete Empirical Interpolation Method* or gappy POD.
- The ultimate goal of ROMs is to apply them to design optimization, uncertainty quantification, safety analysis, and so on.

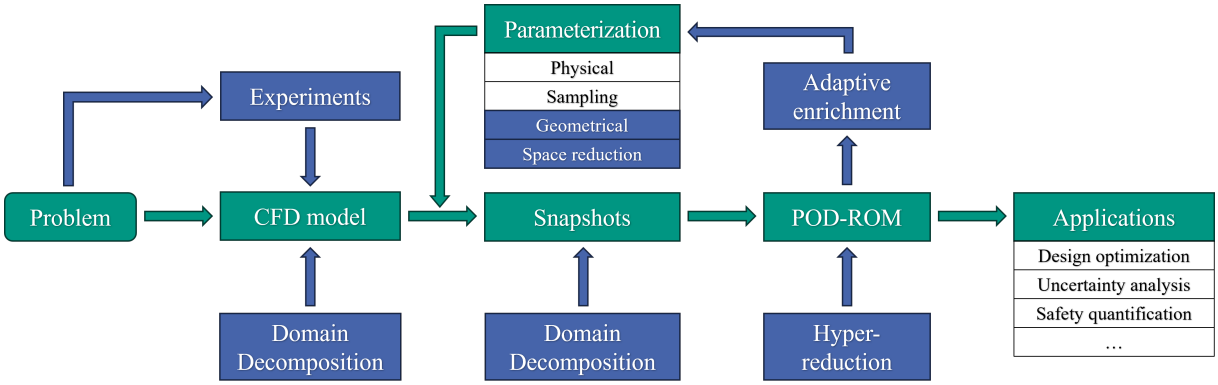


Figure 6.2: A flow chart summarizing comprehensive best-practice guidelines for implementation and adaptation of ROMs for practical applications. CFD models are verified and validated either experimentally or against benchmark data. ROMs can be developed using an intrusive or a non-intrusive approach, and then verified by comparison with high-fidelity CFD solutions. *Domain decomposition* is incorporated either before or after FOM calculations are performed. *Adaptive enrichment* can optimally add new snapshots if needed. *Hyper-reduction* can be employed to further reduce the computational cost of ROMs. Once constructed, ROMs can be used in various scenarios.

A Traditional approaches and review

A.1 Numerical methods for nuclear engineering

A.1.1 Lumped parameter models

System codes were used to analyze and design reactors since the 1960s [388] and are still considered necessary tools nowadays. More descriptions regarding their fundamental principles, development, and current status are summarized in the papers [389; 390]. As shown in Fig. A.1, the reactor system, including the core, pipelines, and heat exchangers, is roughly modeled by a set of 1D control volumes. The tools can investigate system-level dynamics across various scenarios. However, due to these simplifications, many empirical *ad hoc* correlations based on experiments are required to complement the conservation equations, such as those for flow resistance, wall-fluid heat transfer, boiling processes, and two-phase flow. Keep in mind that the empirical relationships are limited to specific reactors. Thus, engineers must conduct additional experiments for new designs, which is cost-ineffective and time-consuming [13].

The *Subchannel codes* are developed to specifically model flow and heat transfer in the core. It was proposed around the 1970s [16], and utilized finer discretization. The division of subchannels of a 37-pin rod bundle is sketched in Fig. A.2. Note that the transverse fluid exchange between parallel channels is roughly correlated with the axial flow. The approach can

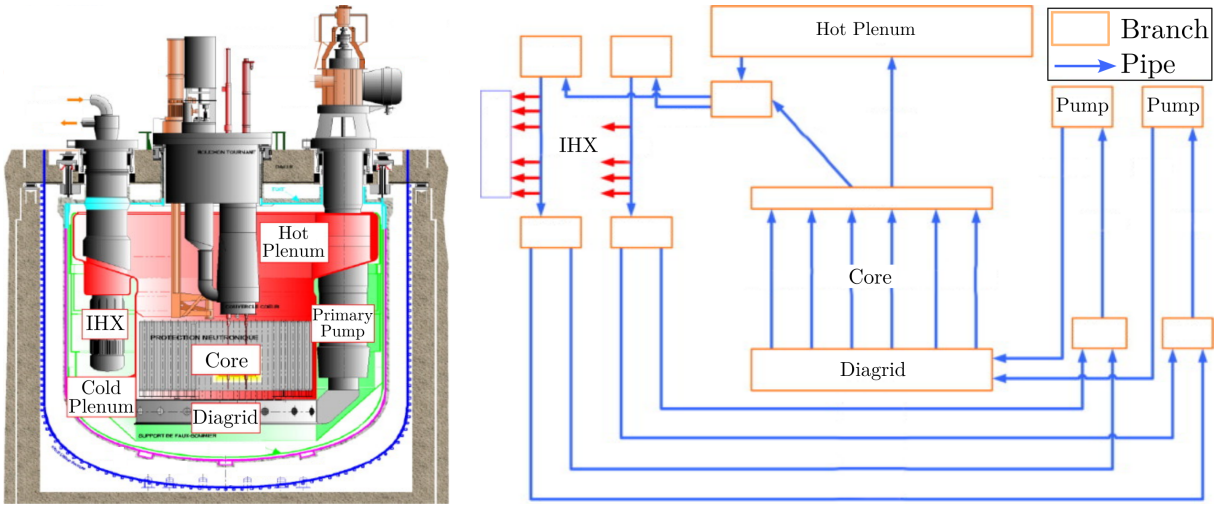


Figure A.1: Overview of a liquid sodium cooled fast reactor (left) and its discretization by a 1D system code (right). The components are modeled with coarse control volumes. Figures taken with permission from [18], copyright owned by Elsevier.

predict the velocity and temperature distributions in both steady-state and transient conditions at minimal computational cost. However, as in system codes, they also require various experiment-based correlations to account for essential phenomena. That means a massive investment and prolonged experiments are unavoidable [391].

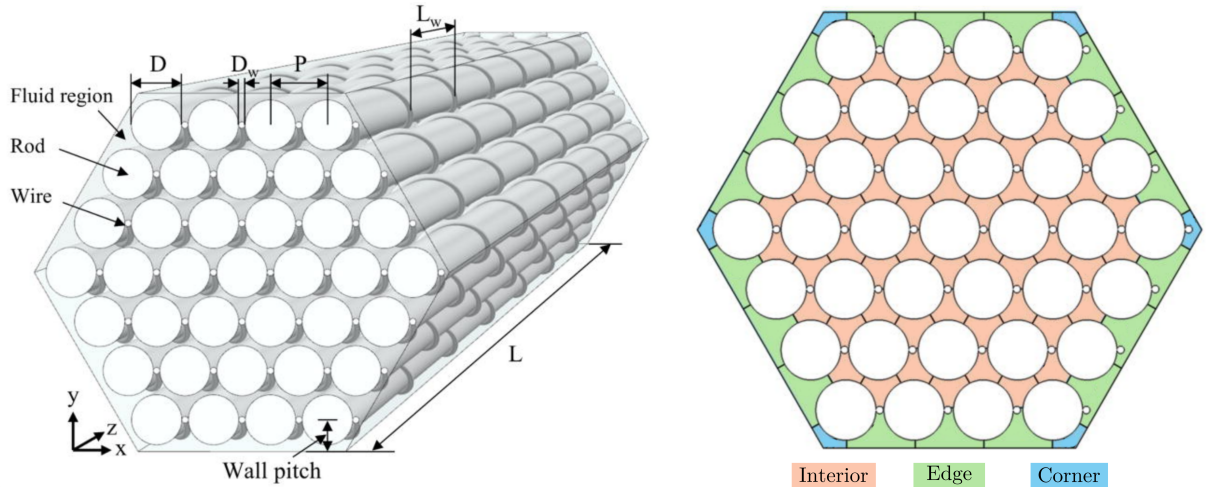


Figure A.2: Cross-section of a 37-pin rod bundle (left) and its division (right). The model comprises three subchannel types: interior, edge, and corner. Figures taken with permission from [20], copyright owned by AIP Publishing.

A.1.2 CFD and turbulence modeling

A.1.2.1 Reynolds-averaged Navier–Stokes

The RANS denotes time-averaged Navier–Stokes equations. The instantaneous quantity is decomposed into a time-averaged and a fluctuating part, referred to as the *Reynolds decomposition*. The RANS system is formulated based on mean flow equations, which introduce some additional unknowns, e.g., the *Reynolds stresses*. *Turbulence models* are developed to add extra equations to close the system. They are based on a presumption suggested by Boussinesq, in which Reynolds stresses are analogous to viscous stresses. Note that these simplified methods involve several constants that need to be adjusted through experiments, and additional treatments or correlations are required to account for the boundary layer.

RANS-based CFD consumes a considerable amount of computational resources to approximate turbulence, resulting in widespread use for industrial geometries. It has been intensively utilized for the numerical analysis of LMFRs. The RANS-based simulations for rod bundles involve various conditions (see Fig. A.3), including thermal-hydraulic assessments [56; 57; 58], flow blockages [59; 60; 61], rod deformation [62; 63; 64], etc. Moreover, RANS can incorporate some simplification techniques (e.g., the porous media method) to model the thermal-hydraulic behaviour of large-scale components in nuclear systems, including the entire reactor vessel [392; 393]. Studies that address uncertainty are also observed [67; 394; 15]. Actually, the

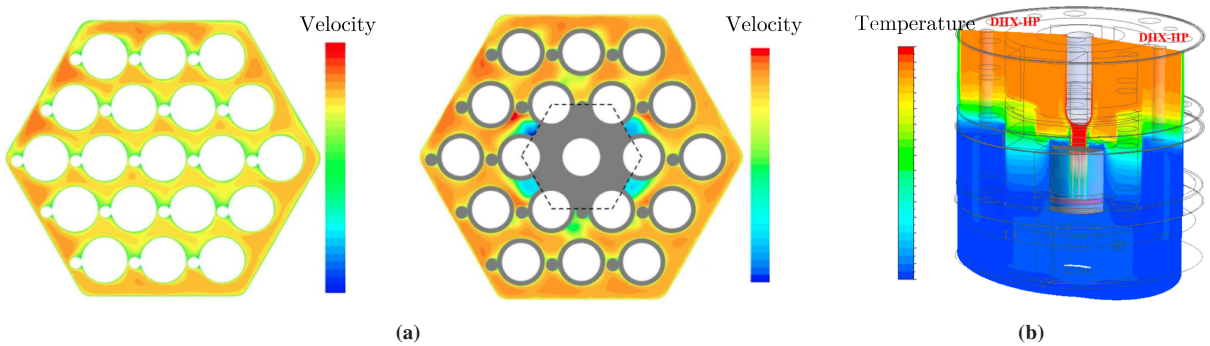


Figure A.3: RANS results for LMFRs. (a) Velocity profiles in a 19-pin wire-wrapped (left) and a blocked bundle (right). Figures taken with permission [398], copyright owned by Elsevier. (b) RANS-based CFD employing porous media method for analysis of the whole reactor. Figure taken with permission from [392], copyright owned by Elsevier.

RANS results are validated by experiment databases with acceptable accuracy, as indicated in [395; 396; 397].

In this research, RANS-based CFD is employed to conduct a series of simulations. The equations, turbulence models, FVM discretization, solver, and other relevant details are illustrated in Sec. 3.1.

The RANS describes the non-dynamic and time-averaged behavior of all turbulence eddies. However, investigations have already shown that large eddies are anisotropic, whereas the behavior of small eddies is isotropic and universal. Hence, the mean model is not sufficient to represent the phenomena.

A.1.2.2 Large eddy simulation

The LES appeared to meet the requirement for capturing coherent turbulent structures. LES employs a spatial filter to retain the large-scale eddies, and the remaining stresses, commonly termed the *Sub-grid-scale* (SGS) stresses, are approximated using a turbulence model. Smagorinsky proposed the basic concepts of this model following Boussinesq's ideology, whereby local SGS stresses are defined as proportional to the local strain rate of the resolved flow.

LES provides a better representation of turbulent features than RANS, but it requires significantly finer spatial and temporal resolutions to capture eddy motion, resulting in a considerable increase in simulation costs.

With the rapid development of computational resources over the past decade, LES has been applied to simulate the fluid dynamics in nuclear engineering [55]. Also, LES results are adopted to validate RANS simulations [37].

It is worth reminding that the open-source code *Nek5000* was the most commonly used LES software for LMFRs. Through the references, Nek5000 managed to create LES models for LMFRs in different scales, including thermal-hydraulic of a single or several subchannels [65;

66]. Investigations of the fuel rods focus primarily on flow characteristics, while heat transport is often overlooked due to computational limitations. Argonne National Laboratory has published several studies on wire-wrapped rod bundles of different sizes. These includes bundles with the rod number of 7 [67], 19 [68], 61 [69], and 217 [70]. In addition to the standard bundle, Nek5000 is also employed to simulate a 61-pin assembly that contains bent rods [399]. The velocity profiles of several simulations are presented in Fig. A.4. It is easy to see that LES better captures local-scale features than RANS, while RANS results are numerically averaged (see Fig. A.3).

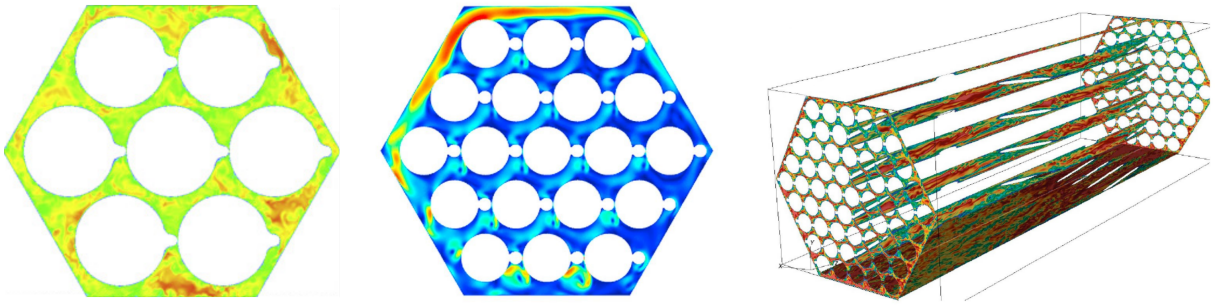


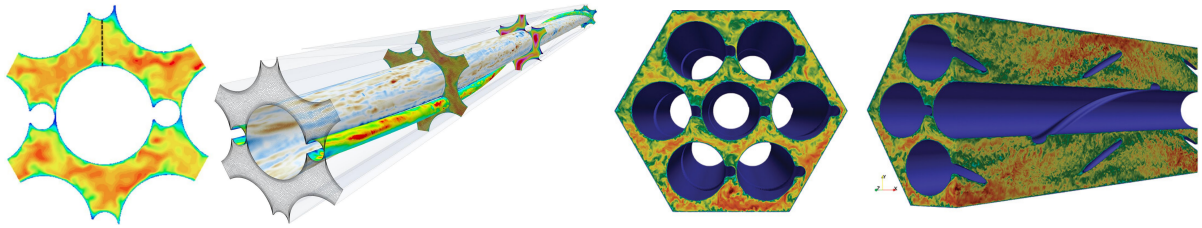
Figure A.4: LES simulations for 7-pin (left), 37-pin (middle), 61-pin (right) rod bundles. Figures taken with permission from [67; 68; 69], copyright owned by Elsevier.

Two overview documents [55; 400] are suggested for readers who are concerned with the application of LES in nuclear engineering.

A.1.2.3 Direct numerical simulation

In contrast to the two methods above, which integrate empirical turbulence models, DNS is a fully resolved method. It uses a sufficiently fine spatial mesh and small time steps, and consequently, the smallest eddies and the fastest fluctuations can be captured in a transient solution. It is generally accepted that DNS is time-consuming but very accurate. It can be used to validate and evaluate RANS and LES turbulence models [36].

Be aware that there are a few publications that apply DNS to nuclear thermal hydraulics. Readers can find the overview discussed in [36]. Emilio Baglietto et al. exploited DNS to analyze flow behaviour in a minimum symmetric geometry even smaller than a subchannel [71]. In addition to fluid dynamics, Diego Angeli et al [72], and Ferry Roelofs et al. [73] also investigated heat transport in a single inner subchannel. A simulation for a larger computational domain is reported by Aaque Shams et al and Daniele Dovizio, [74; 37]. They managed to compute the flow and thermal field for the region around a wire-wrapped rod. Two more recent articles by Bourdot Dutra et al. presented a DNS including heat transfer in a seven-pin wire-wrapped bundle [75]. The plots in Fig. A.5 indicate that DNS can well represent small-scale flow variances. Velocity fields exhibit greater complexity and richer dynamics than averaged RANS solutions and filtered LES simulations.



(a) Subchannels around a rod

(b) Seven-pin bundle

Figure A.5: DNS velocity fields for rod bundles. (a) Figures taken with permission from [73], copyright owned by Elsevier. (b) Figure taken with permission from [75], copyright owned by Taylor & Francis.

A.1.2.4 Discussions

Now is the time to describe the differences among the three techniques and summarize their potential in nuclear engineering.

Based on the references cited above, the implementation of the three techniques for LMFRs can be summarized as follows. RANS can be applied to various physical scenarios and large-scale geometries, e.g., fuel assemblies, or even the whole reactor core. Recently, advances in hardware have enabled the implementation of LES to simulate multiple rod bundles. Over the past several years, several DNS studies have been published, which consider only smaller geometries, such as subchannels and a seven-pin wire-wrapped bundle. Given the need for simulations in LMFRs and the complexities of nuclear engineering applications, RANS remains the most widely used numerical method for thermal-hydraulic analysis of reactors [37].

Additionally, the competencies of RANS, LES, and DNS for turbulent modeling varied from the theoretical perspective, as illustrated in Fig. A.6. Actually, turbulence involves multiscale eddies with different sizes and energy levels. The relation is illustrated as the variation of the eddy's *energy-spectrum function* $E(\kappa)$ against the eddy length scales(i.e., wave number κ). More instructions about the knowledge can be found in the book [22]. The comparison also indicates that the RANS can only approximate the averaged field. LES represents large-scale eddies and hence captures the overall dynamics. However, the detailed and precise characteristics can be resolved more precisely by DNS.

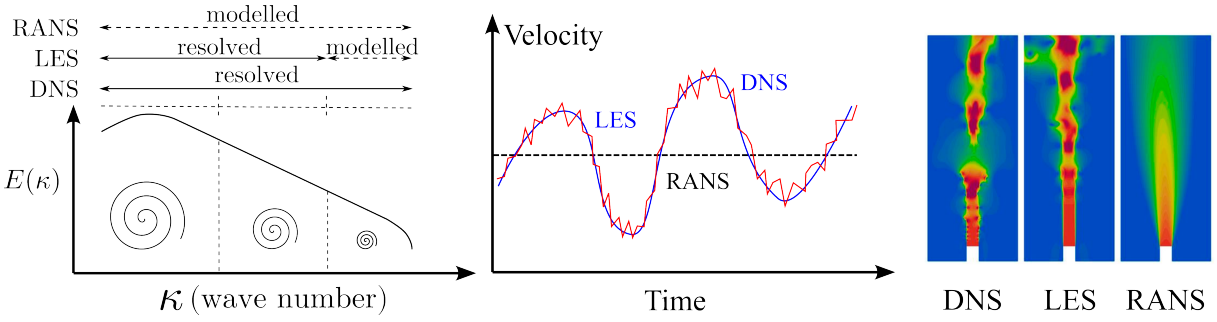


Figure A.6: A comparison between RANS, LES, and DNS for turbulent simulations. Left: The variation of the eddy's *energy-spectrum function* $E(\kappa)$ against the wave number κ. Middle: A sketch of the capability to capture turbulent variances. Right: Results of the jet flow simulation using the three methods. Figure on the left is taken with permission from [401], copyright owned by MDPI.

Obviously, DNS and LES can yield more accurate estimates of turbulent variance. However, the improvement usually requires an incredible amount of computational power. The geometric complexities and multiscale features of reactor components make it extremely expensive to generate a high-quality mesh for them. Furthermore, the exceptionally high computational costs limit such simulations to well-funded institutions such as Argonne National Laboratory [69; 70] or to projects supported by the U.S. Department of Energy [75]. Indeed, DNS and LES are impractical for most research groups. Consequently, considering the importance of parametric solutions, RANS-based CFD is still an essential tool for the near and foreseeable future.

B Methodology and theory

Note that the methodologies and theories applied in this research are introduced in Chapter 3. This appendix provides more details regarding some specific techniques:

- Explanations for high-fidelity CFD, including near-wall treatment, FVM, and OpenFOAM.
- Computational procedures of POD.
- Additional relevant formulations of the three regression models: RBF, GPR, and ANN.

B.1 High fidelity CFD analysis

Reynolds-averaged Navier-Stokes and the SST $k-\omega$ turbulence model are already introduced in Section 3.1.1. The following sections present more details regarding near wall treatment of the turbulence model, the finite volume method, and OpenFOAM.

B.1.1 Near-wall treatment

According to boundary layer theory, the velocity varies significantly within the boundary layer [52]. Therefore, accurate near-wall treatment is essential for reliable RANS simulations [23].

For turbulent flows, the variance of the mean velocity profile near a solid boundary (in the direction normal to the boundary), $\overline{\mathbf{u}} \cdot \mathbf{n}$, can be described by the *law of the wall*, as shown in Fig. B.1. The curve is plotted regarding non-dimensional velocity u^+ and y^+, which are defined as

$$y^+ = \frac{y u_\tau}{\nu}, \quad u^+ = \frac{u}{u_\tau},$$

with

$$u_\tau = \sqrt{\frac{\tau_w}{\rho}}, \quad \tau_w = \rho\nu \left.\frac{\partial u_x}{\partial y}\right|_{y=0},$$

where y is the normal distance from the wall, τ_w is the wall shear stress, and u_τ is called the friction velocity or shear velocity.

The red line (see Fig. B.1) denotes results extracted from DNS simulations. Considering the profile, the boundary layer is divided into three regions, in which u^+ and y^+ can be correlated as:

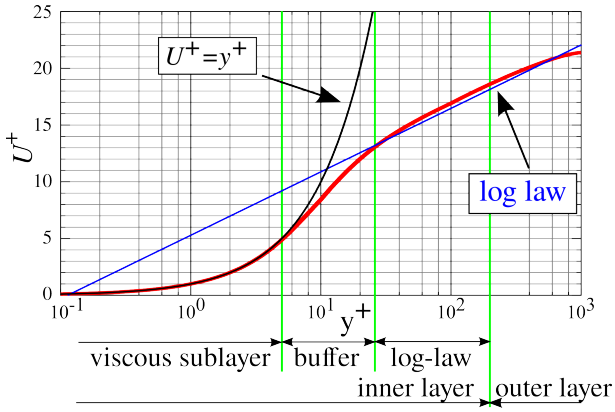


Figure B.1: Law of the wall for Dimensionless mean velocity u^+ profile against dimensionless wall distance y^+. Inner layers: viscous sublayer ($y^+ < 5$), buffer region ($5 < y^+ < 30$), and log-law area ($y^+ > 30$). Red line: DNS results. Figure is taken with permission from [402], copywriter owned by the Wikimedia Foundation, Inc.

1. *Viscous sublayer* $(y+ < 5)$ with linear relation, $u^+ = y+$.
2. *Buffer layer* $(5 < y+ < 30)$, transitional region without any simple analytical function.
3. *Logarithmic layer* $(y+ > 30)$ with $u^+ = \frac{1}{\kappa} \ln y^+ + C^+$, where κ is Von Kármán constant and C^+ is a constant.

The physical features of the layers are different. The *viscous sublayer* is characterized by laminar flow and is dominated by the viscous effect. The *buffer layer* is simultaneously governed by molecular viscosity and turbulence. The turbulence plays a major role in the *log-law* region. The three layers are also known as the *inner layer*. The *outer layer* refers to the y^+ range where the profile does not perfectly satisfy the log-law. The exact upper limit depends on the Reynolds number. Typically, it is around $y^+ > 200$.

Considering the velocity distribution, two approaches are typically applied to model the region, as sketched in Fig. B.2. In one approach, a fine mesh is generated along the normal to the wall, allowing turbulence models to resolve the large velocity gradients within viscosity-affected regions (the viscous sublayer and buffer layer). For the other one, the flow is represented by semi-empirical correlations called "wall functions" that account for the law of the wall. Therefore, a single layer of grids can cover the $y^+ < 30$ region. See [22; 23] for details.

Note that the former is accurate but expensive, and it is pertinent to the $k - \omega$ and SST $k - \omega$ models for resolving the boundary layer. In contrast, the latter is essential for the $k - \epsilon$ model to roughly represent near-wall flow and is feasible and economical for engineering practice.

In this research, the SST $k - \omega$ model with a finer mesh is adopted for FOM simulations.

B.1.2 Finite Volume method and OpenFOAM

As the open-source CFD software, *Open Field Operation And Manipulation* (OpenFOAM) is applied for performing FOM calculations [349; 350; 337; 351]. The RANS solver and its

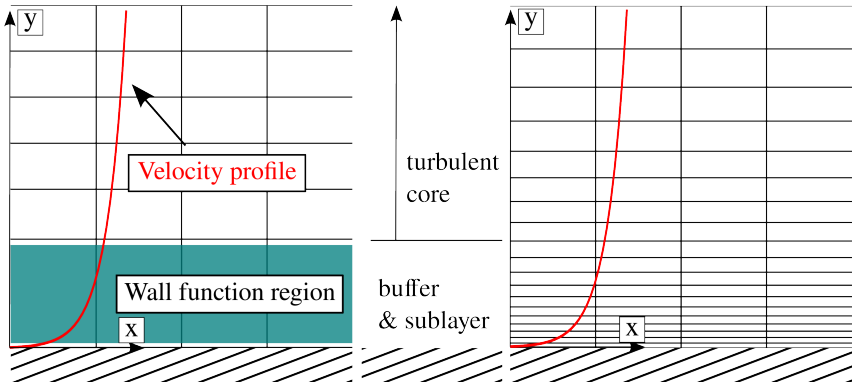


Figure B.2: Wall treatment using wall function (left) or near-wall modelling (right) approaches. Wall functions are applied to a high-Reynolds-number turbulent model. The low-Reynolds-number model should be employed to resolve the inner region.

basis, *Finite Volume Method*, are briefly introduced. The book [24] is highly recommended for interested readers.

B.1.2.1 Finite Volume method

The FVM is one of the most widely used discretization techniques in CFD. It discretizes the computational domain into numerous control volumes, and surface (2D) or volumetric (3D) integrals are used to construct the conservative formulation of the governing PDEs in each volume. The conservation of variables in the frame of FVM for a 2D cell is sketched in Fig. B.3. The value of any variable is stored at the centroid C for each cell, with a single quantity per cell. The conservation of the PDE in the cell depends on the flux across the interface f and the source.

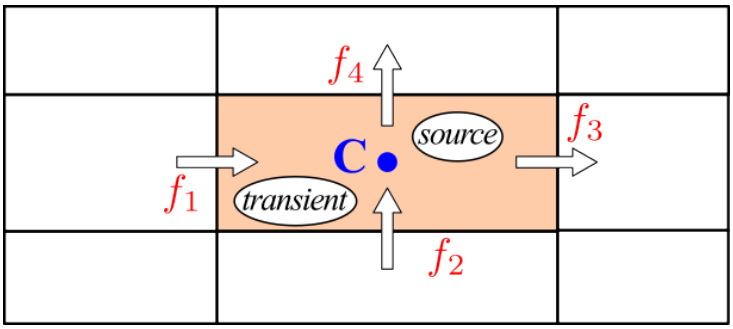


Figure B.3: Conservative description of a 2D cell (marked in peach) in the framework of FVM. The values of all variables are stored in the centroid **C**. The balance is achieved considering the transient variance, the interior source, and transports from neighbours (e.g., $f_1, \ldots, f_4$).

Different terms in a PDE can be computed in the configuration. Given the cell V_C and its outer surface area vector $\mathbf{S} = S\mathbf{n}$, $\int_{V_C} \nabla \cdot \mathbf{u}$ can be computed through the divergence theorem, and it becomes $\int_{\partial V_C} \mathbf{u} \cdot \mathbf{n} = \sum_{f \in \partial V_C} \mathbf{u}_f \cdot \mathbf{n}_f S_f = \sum_{f \in \partial V_C} \mathbf{u}_f \cdot \mathbf{S}_f$. Similarly, the gradient term can be rewritten as $\int_{V_C} \nabla p = \int_{\partial V_C} p\mathbf{n} = \sum_{f \in \partial V_C} p_f \cdot \mathbf{S}_f$.

The above expressions are computed via inner face values. As the FVM quantities are located at the centroid, specific algorithms, referred to as *numerical schemes*, should be adopted to compute the face values from the cell values. Only a single value is defined in each $f \in \partial V_C$. The accuracy of FVM solutions is highly affected by the schemes [24].

By applying the FV integral operation, equation (3.3) for a cell V_C is formulated as

$$\begin{aligned} \int_{V_C} \frac{\partial \mathbf{u}}{\partial t} + \int_{V_C} (\mathbf{u} \cdot \nabla)\mathbf{u} = -\int_{V_C} \nabla p + \int_{V_C} \nabla \cdot \left[\nu_{\text{eff}} \left(\nabla \mathbf{u} + \nabla \mathbf{u}^{\mathrm{T}}\right)\right] + \int_{V_C} \mathbf{f}, \\ \int_{V_C} \nabla \cdot \mathbf{u} = 0, \end{aligned} \tag{B.1}$$

where the effective viscosity $\nu_{\text{eff}} = \nu + \nu_t$. Note that the time-averaged notation is neglected for simplicity.

In this dissertation, the steady-state RANS without a source term is adopted, that is

$$\begin{aligned} \int_{V_C} (\mathbf{u} \cdot \nabla)\mathbf{u} = -\int_{V_C} \nabla p + \int_{V_C} \nabla \cdot \left[\nu_{\text{eff}} \left(\nabla \mathbf{u} + \nabla \mathbf{u}^{\mathrm{T}}\right)\right] \\ \int_{V_C} \nabla \cdot \mathbf{u} = 0, \end{aligned} \tag{B.2}$$

The computation of various terms in the FVM framework is illustrated. The *convection* term is given by

$$\begin{aligned} \int_{V_C} (\mathbf{u} \cdot \nabla)\mathbf{u} &= \int_{V_C} \nabla \cdot (\mathbf{u}\mathbf{u}) \\ &= \int_{\partial V_C} (\mathbf{u} \cdot \mathbf{n})\mathbf{u} \\ &= \sum_{f \in \partial V_C} \mathbf{u}_f \cdot \mathbf{n}_f S_f \mathbf{u}_f \\ &= \sum_{f \in \partial V_C} \mathbf{u}_f \cdot \mathbf{S}_f \mathbf{u}_f \\ &= \sum_{f \in \partial V_C} F_f \mathbf{u}_f, \end{aligned}$$

where $F_f = \mathbf{u}_f \cdot \mathbf{S}_f$ is the flux through f.

The effective *diffusion* (also known as *Laplacian*) term is computed as

$$\int_{V_C} \nabla \cdot \left[\nu_{\text{eff}} \left(\nabla \mathbf{u} + \nabla \mathbf{u}^{\mathrm{T}}\right)\right] = \sum_{f \in \partial V_C} \nu_{\text{eff}} \left(\nabla \mathbf{u}_f + \nabla \mathbf{u}_f^{\mathrm{T}}\right) \cdot \mathbf{S}_f.$$

The other terms can be obtained in the similar manner, namely transient $\int_{V_C} \frac{\partial \mathbf{u}}{\partial t} = \frac{\partial \mathbf{u}}{\partial t} V_C$, pressure gradient $\int_{V_C} \nabla p = \sum_{f \in \partial V_C} \mathbf{S}_f p_f$, and source $\int_{V_C} \mathbf{f} = \mathbf{f} V_C$. The procedure can be applied to the k and ω equations accordingly. The boundary conditions can be easily enforced

because the surface values are already included in each term, which is not covered here. Remark that distinct numerical schemes might be applied for each term. More details are presented in [24; 337; 351].

The numerical schemes adopted for this dissertation are listed

1. Gauss theorem for the gradient computation, as mentioned above.
2. Linear-upwind as the divergence scheme (e.g., convection term) [403].
3. Linear interpolation for computing Laplacian (e.g., diffusion) terms.

B.1.2.2 Meshing

Spatial discretization of a computational domain will result in either structured or unstructured meshes [24], as illustrated by Fig. B.4.

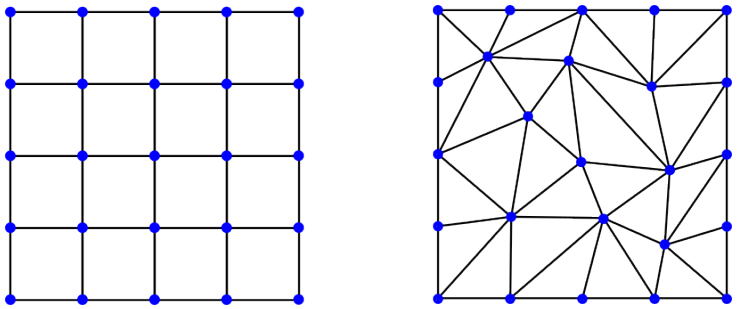


Figure B.4: Structured (left) and unstructured (right) mesh with the same number of vertices. All interior vertices of the former are connected with four neighbours. The connection varies from five to seven for the unstructured. Redraw based on [404].

The former stores the cells in a structured order, which is typically compared by quadrilaterals or hexahedra in two and three dimensions, respectively. It can produce high-quality mesh, require little storage space, and be numerically simpler due to its well-organized features.

However, a structured grid system is hardly feasible for complex domains, especially in industrial applications with multiscale geometries. The unstructured approach is proposed because it can handle more common scenarios. However, this flexibility introduces additional complexity in creating the topology across all cells [24].

The geometries and meshing varied due to the problem configurations. Thus, a mesh-independent study is required to ensure that mesh sizes do not significantly influence the simulation results. Note that simulation accuracy typically depends on the mesh and numerical schemes.

In this research, structured meshes are employed for simple cases, while unstructured meshes are utilized for complex geometries.

B.1.2.3 SimpleFoam: an incompressible solver

Once the mesh is available, the integral PDEs for all cells, along with the boundary conditions, can be assembled into a global system, which is subsequently solved iteratively. As RANS contains several equations, including non-linear terms, specific algorithms should be applied to achieve the linearization and effective iterations of the problem. For incompressible steady state flow, the most widely used algorithm is called *Semi-Implicit Method for Pressure-Linked Equations* (SIMPLE), which was developed by Brian Spalding and Suhas Patankar [352; 353]. Note that, in the framework, the *pressure Poisson equation* is adopted to substitute the continuity equation. The steps and implementations are detailed in [405; 23; 24].

The SIMPLE algorithm is employed to create *simpleFoam*, an incompressible steady-state solver for both laminar and turbulent (RANS) flow. See [337; 24] for a more in-depth explanation.

The solving of the discretized and linearized equations is performed iteratively. Various techniques have been proposed to solve these systems, such as *Gauss-Seidel*, *Preconditioned Biconjugate Gradient*, *Geometric Algebraic MultiGrid* methods, etc. [337]. Typically, the solving approaches do not affect the solution accuracy once the problem has converged. Whereas they might significantly affect the convergence rate and, consequently, influence computation time.

For this study, the simpleFoam with RANS is utilized to generate high-fidelity solutions. To balance convergence rate and computational cost, the simulations adopted the *Preconditioned Biconjugate Gradient* and *Geometric Algebraic MultiGrid* algorithms to solve the algebraic systems.

B.2 Algorithms for Proper Orthogonal Decomposition

POD can be computed with two equivalent methods, namely *Singular Value Decomposition* (SVD) and *Eigendecomposition* of the correlation matrix. Their formulations and procedures are presented below.

B.2.1 Singular Value Decomposition

Given the snapshot matrix $\mathbf{S}$ with shape $N_h \times N_\mu$, by applying SVD, one can obtain

$$\mathbf{S} = \tilde{\mathbf{\Phi}}\tilde{\mathbf{\Sigma}}\tilde{\mathbf{\Psi}}^{\mathrm{T}},$$

where $\tilde{\mathbf{\Phi}} \in \mathbb{R}^{N_h \times N_h}$ and $\tilde{\mathbf{\Psi}} \in \mathbb{R}^{N_\mu \times N_\mu}$ are orthonormal singular matrices. Moreover,

$$\tilde{\mathbf{\Sigma}} = \mathrm{diag}\left\{\sigma_1, \ldots, \sigma_{N_\mu}\right\} \in \mathbb{R}^{N_h \times N_\mu},$$

and the singular values satisfy $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_{N_\mu} \geq 0$. Each column vector of $\tilde{\mathbf{\Phi}}$ can be regarded as a POD mode. The POD coefficients can be computed from $\tilde{\mathbf{A}}^{\mathrm{T}} = \tilde{\mathbf{\Sigma}}\tilde{\mathbf{\Psi}}^{\mathrm{T}}$. Thus, $\mathbf{S} = \tilde{\mathbf{\Phi}}\tilde{\mathbf{A}}^{T}$.

B.2.2 Eigendecomposition

The correlation matrix of $\mathbf{S}$ is defined as $\mathbf{C} = \mathbf{S}^{T}\mathbf{S}$. Then, the eigenvalue decomposition of $\mathbf{C}$ results

$$\mathbf{C}\mathbf{V} = \mathbf{V}\boldsymbol{\lambda}, \tag{B.3}$$

where $\mathbf{V}$ contains eigenvectors and $\boldsymbol{\lambda} = \mathrm{diag}\{\lambda_1, \ldots, \lambda_{N_\mu}\}$ is a diagonal matrix consists of eigenvalues with $\lambda_1 \geq \cdots \geq \lambda_{N_\mu} \geq 0$.

The POD mode and coefficient matrices are obtained with

$$\tilde{\mathbf{\Phi}} = \frac{1}{\sqrt{\lambda_i}}\mathbf{S}\mathbf{V}, \quad \tilde{\mathbf{A}}^{\mathrm{T}} = \sqrt{\lambda_i}\mathbf{V}^{\mathrm{T}}.$$

Actually, the two methods are equivalent. $\mathbf{C}$ can be expressed as

$$\begin{aligned} \mathbf{C} &= \tilde{\mathbf{\Psi}}\tilde{\mathbf{\Sigma}}^{\mathrm{T}}\tilde{\mathbf{\Phi}}^{\mathrm{T}}\tilde{\mathbf{\Phi}}\tilde{\mathbf{\Sigma}}\tilde{\mathbf{\Psi}}^{\mathrm{T}} \\ &= \tilde{\mathbf{\Psi}}\tilde{\mathbf{\Sigma}}^{\mathrm{T}}\tilde{\mathbf{\Sigma}}\tilde{\mathbf{\Psi}}^{\mathrm{T}}, \end{aligned} \qquad \mathbf{C}\tilde{\mathbf{\Psi}} = \tilde{\mathbf{\Psi}}\tilde{\mathbf{\Sigma}}^{\mathrm{T}}\tilde{\mathbf{\Sigma}}. \tag{B.4}$$

Considering equations (B.3) and (B.4), the equalities hold $\tilde{\mathbf{\Psi}} = \mathbf{V}$ and $\tilde{\mathbf{\Sigma}}^{\mathrm{T}}\tilde{\mathbf{\Sigma}} = \boldsymbol{\lambda}$. The latter can be further written as $\lambda_i = \sigma_i^2$. Thus, $\tilde{\mathbf{\Phi}} = \mathbf{S}\tilde{\mathbf{\Psi}}\boldsymbol{\lambda}^{-1/2} = \mathbf{S}\mathbf{V}\boldsymbol{\lambda}^{-1/2}$ and $\tilde{\mathbf{A}}^{\mathrm{T}} = \tilde{\mathbf{\Sigma}}\tilde{\mathbf{\Psi}}^{\mathrm{T}} = \boldsymbol{\lambda}^{1/2}\mathbf{V}^{\mathrm{T}}$.

Note that the relative projection error for n-rank POD can be expressed in terms of the singular values as follows

$$\mathcal{E}_n = \frac{\sqrt{\sum_{i=n+1}^{N_\mu} \sigma_i^2}}{\sqrt{\sum_{i=1}^{N_\mu} \sigma_i^2}}.$$

B.3 Regression algorithms

B.3.1 Radial Basis Function

Details of RBF are presented here.

For N_μ parameters, it results in a matrix system

$$\begin{bmatrix} \varphi(\|\mu_1 - \mu_1\|) & \cdots & \varphi(\|\mu_1 - \mu_{N_\mu}\|) \\ \vdots & \ddots & \vdots \\ \varphi(\|\mu_{N_\mu} - \mu_1\|) & \cdots & \varphi(\|\mu_{N_\mu} - \mu_{N_\mu}\|) \end{bmatrix} \begin{bmatrix} w_1 \\ \vdots \\ w_{N_\mu} \end{bmatrix} = \begin{bmatrix} \boldsymbol{\alpha}(\mu_1) \\ \vdots \\ \boldsymbol{\alpha}(\mu_{N_\mu}) \end{bmatrix},$$

where the unknowns of w_j are computed by solving the matrix equation. Then, for any μ_{new}, the corresponding POD coefficients are $\boldsymbol{\alpha}(\mu_{\text{new}}) = \sum_{j=1}^{N_\mu} w_j \varphi(\|\mu_{\text{new}} - \mu_j\|)$.

Various definitions of the RBF kernel φ have been proposed. Let us define $r = \|\mu - \mu_j\|$, and $\varphi(r) = \varphi(\|\mu - \mu_j\|)$. Several commonly used radial basis functions include [373],

- Gaussian
$$\varphi(r) = e^{-(\varepsilon r)^2}.$$
- Polyharmonic spline
$$\begin{aligned} \varphi(r) &= r^k, && k = 1, 3, 5, \ldots \\ \varphi(r) &= r^k \ln(r), && k = 2, 4, 6, \ldots \end{aligned}$$
Several common cases are
  - Thin plate spline
$$\varphi(r) = r^2 \ln(r).$$
  - Cubic
$$\varphi(r) = r^3.$$
  - Quintic
$$\varphi(r) = -r^5.$$
- Inverse quadratic
$$\varphi(r) = \frac{1}{1 + (\varepsilon r)^2}.$$
- Multiquadric or inverse multiquadric
$$\begin{aligned} \varphi(r) &= \sqrt{1 + (\varepsilon r)^2} \\ \varphi(r) &= \frac{1}{\sqrt{1 + (\varepsilon r)^2}}. \end{aligned}$$

In the above formulation, ε is a parameter that must be tuned via numerical tests.

When the polynomial term is added, the RBF interpolation becomes

$$\begin{bmatrix} \varphi(\|\mu_1-\mu_1\|) & \cdots & \varphi(\|\mu_1-\mu_{N_\mu}\|) & 1 & \cdots & \mu_1^{n-1} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ \varphi(\|\mu_{N_\mu}-\mu_1\|) & \dots & \varphi(\|\mu_{N_\mu}-\mu_{N_\mu}\|) & 1 & \cdots & \mu_{N_\mu}^{n-1} \\ 1 & \cdots & 1 & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ \mu_1^{n-1} & \cdots & \mu_{N_\mu}^{n-1} & 0 & \cdots & 0 \end{bmatrix} \begin{bmatrix} w_1 \\ \vdots \\ w_{N_\mu} \\ c_1 \\ \vdots \\ c_n \end{bmatrix} = \begin{bmatrix} \boldsymbol{\alpha}(\mu_1) \\ \vdots \\ \boldsymbol{\alpha}(\mu_{N_\mu}) \\ 0 \\ \vdots \\ 0 \end{bmatrix},$$

where n is the order of the polynomial, and c_i are additional unknowns.

Note that the polynomial function is required for some kernels to ensure the interpolation problem is well-posed. The recommended order n for *thin plate spline*, *cubic* and *quintic* are $1, 1, 2$, respectively [373].

B.3.2 Gaussian Process Regression

The computation of the kernel matrix for GPR is presented here (see also in [375]). Suppose $\Sigma = \Sigma(\mathbf{X}, \mathbf{X})$ and $\mathbf{X} = \{x_i, i = 1, \dots, n\}$, each element of the kernel matrix is indexed as Σ_{ij}. Several widely used kernel is listed here [375]. Those are:

- RBF kernel

  $$\Sigma_{ij} = \exp\left(-\frac{|x_i - x_j|^2}{2l^2}\right), \quad i, j = 1, \dots, n$$

  where l is a length-scale that needs to be determined and $|x_i - x_j|$ is the Euclidean distance.

- Matern kernel

  $$\Sigma_{ij} = \frac{1}{\Gamma(\nu)2^{\nu-1}}\left(\frac{\sqrt{2\nu}}{l}|x_i - x_j|\right)^{\nu} K_\nu\left(\frac{\sqrt{2\nu}}{l}|x_i - x_j|\right),$$

  where $K_\nu(\cdot)$ is a modified Bessel function and $\Gamma(\cdot)$ is the gamma function. ν is an additional quantity to be optimized.

- Exp-Sine-Squared kernel

  $$\Sigma_{ij} = \exp\left(-\frac{2\sin^2(\pi|x_i - x_j|/p)}{l^2}\right),$$

  where p is the *periodicity* parameter should be assigned.

B.3.3 Artificial Neural Network

Here, a series of widely used activation functions for ANNs is presented. Those are:

- Step Function
$$\sigma(x) = \begin{cases} 0, & x \leq T \\ 1, & x > T \end{cases}$$
, where T is a predefined value.
- Sigmoid
$$\sigma(x) = \frac{1}{1 + e^{-x}}.$$
- Hyperbolic Tangent
$$\sigma(x) = \tanh(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}}.$$
- Basic Rectified Linear Unit
$$\sigma(x) = \max(0, x).$$

C Numerical setups of the FOM simulations

The *numerical schemes* for interpolating inner face values are listed in Table C.1, as well as the accuracy of each term. The choice for each term is mainly decided based on experience.

Table C.1: Numerical schemes.

Term	Discretization scheme	Accuracy
Gradient	Green Gauss theorem	-
Divergence	Linear-upwind	Second order
Laplacian	Linear	First order
Interpolation	Linear	First order

The PDEs are discretized as linear systems and solved iteratively. The solvers are presented in Table C.2. For the momentum (velocity) conservation, k and ω transport equations, the *Preconditioned biconjugate gradient* method is adopted. Two solvers are used to solve the pressure Poisson equation, i.e., *Preconditioned conjugate gradient* and *Geometric algebraic multigrid*. The former is applied to small problems, and the latter is more suitable for large-scale scenarios.

The algebraic system is solved iteratively until the specified convergence criteria are satisfied. The relative residuals for each equation, as defined in [351], are also summarized in Table C.2. For the momentum conservation, the tolerance is set to 10^{-4}, while the rest of the equations are assigned a stricter value of 10^{-5}. The thresholds are applied for all FOM simulations mentioned in this dissertation.

But there is an exception for the residual. In the single-subchannel case, the bulk flow direction is along the z-axis, and the transverse velocity is one or two orders of magnitude lower. It is difficult for the x- and y-axes velocity to reach a residual (relative error) of less than 10^{-4}. Thus, in those calculations, the criterion of the momentum equation is set to 2×10^{-4}.

In addition, the convergence of a CFD simulation does not simply depend on the residual. Several other quantities are also adopted, including plane-averaged pressure, mass flow conservation, and mean wall y^+ values.

Table C.2: Linear system solvers and convergence criteria.

Equation	Solver	Relative residuals
Pressure Poisson equation	*Preconditioned conjugate gradient* *Geometric algebraic multigrid*	10^{-5}
Momentum	*Preconditioned biconjugate gradient*	10^{-4}
k and ω transport	*Preconditioned biconjugate gradient*	10^{-5}

D Global FOMs and ROMs

Several figures and algorithms regarding parameters are listed in this appendix, including those for global and local ROMs.

The aspects that correspond to global approaches are presented as follows. They are listed with respect to three cases shown in Chapter 4.

D.1 A single subchannel

D.1.1 Parameters

Algorithm 3 presents the steps to select *training* P_{train} and *test* P_{test} points from the entire set P.

The size of P is 50, as shown in Fig. D.1. The numbers of P_{train} and P_{test} can be varied to test their influence.

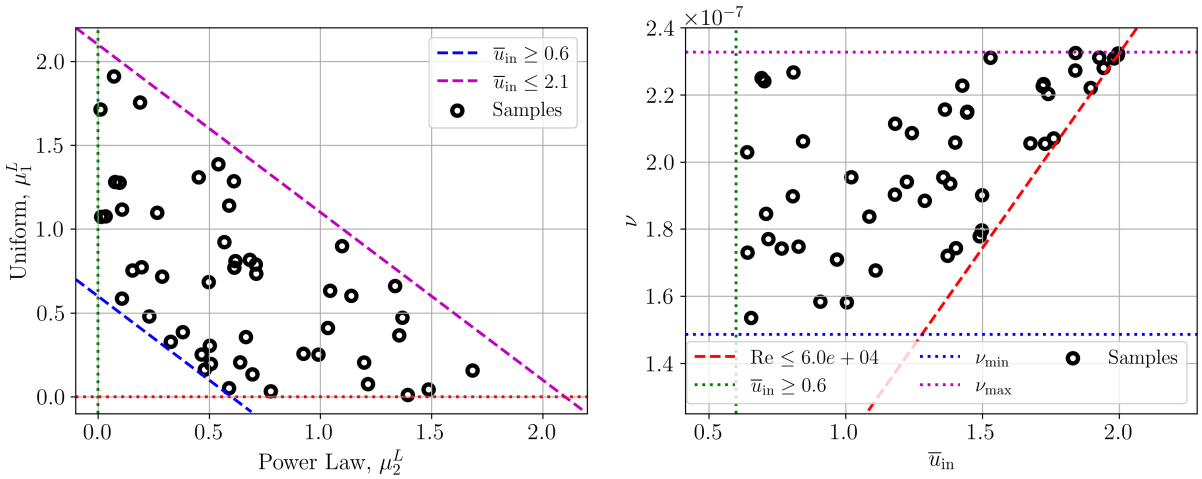


Figure D.1: Selected parameters: ν, μ_1^L, and μ_1^L. The values are chosen to ensure the limits, i.e., $Re \in [1.4, 6.0] \times 10^4$ and $\overline{u}_{\text{in}} \in [0.6, 2.1]$.

For constructing and validating ROMs, 30 points are selected as the training set P_{train} and the remaining 20 as the test set P_{test}. They are plotted in Fig. D.2. To identify the effects of the number of training points, three additional sets with 5, 10, and 20 points are also considered and plotted in the figure.

Algorithm 3 Selection of P_{train} and P_{test} from P

Input: $P = \{\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_{N_\mu}\}$, each $\boldsymbol{\mu}_i = [\nu_i, \mu_{1,i}^L, \mu_{2,i}^L]^{\mathrm{T}}$, number of training points N_{train}
Output: Training set P_{train}, test set P_{test}
Compute the full centroid:

$$\overline{\boldsymbol{\mu}} = \frac{1}{N_\mu} \sum_{\boldsymbol{\mu} \in P} \boldsymbol{\mu}$$

Select the first training point:

$$\boldsymbol{\mu}^{(1)} = \arg\max_{\boldsymbol{\mu} \in P} \|\boldsymbol{\mu} - \overline{\boldsymbol{\mu}}\|_{L^2}$$

Initialize $P_{\text{train}} \leftarrow \{\boldsymbol{\mu}^{(1)}\}$
for $k = 2$ to N_{train} **do**
    Compute the centroid of current P_{train}:

$$\overline{\boldsymbol{\mu}}_{\text{train}} = \frac{1}{|P_{\text{train}}|} \sum_{\boldsymbol{\mu} \in P_{\text{train}}} \boldsymbol{\mu}$$

    Select:

$$\boldsymbol{\mu}^{(k)} = \arg\max_{\boldsymbol{\mu} \in P \setminus P_{\text{train}}} \|\boldsymbol{\mu} - \overline{\boldsymbol{\mu}}_{\text{train}}\|_{L^2}$$

    Update $P_{\text{train}} \leftarrow P_{\text{train}} \cup \{\boldsymbol{\mu}^{(k)}\}$
end for
Set $P_{\text{test}} \leftarrow P \setminus P_{\text{train}}$

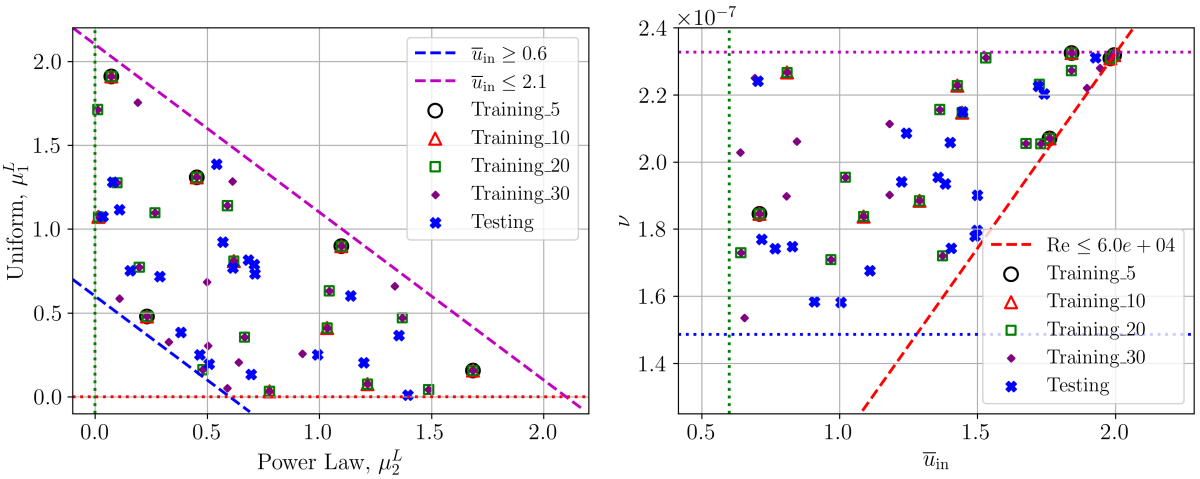


Figure D.2: Train sets with 5, 10, 20, and 30 points for the construction of ROMs for the single subchannel cases. The training sets are overlapped due to the selection algorithm.

D.1.2 Settings of neural networks

The configurations of ANN for global ROMs are summarized in Table D.1. The choices are mainly based on experience and trial-and-error. The framework is *PyTorch* [372], which is a widely used open-source machine learning package.

Table D.1: Configurations of ANN for global ROMs.

Hyperparameter	Value
Loss function	Mean squared error
Optimization algorithm	Adam
Learning rate	0.001
Number of epochs	20000
Normalization	Min-max scaling
Framework	PyTorch

D.2 A bare rod bundle

D.2.1 Parameters

The *uniform* and *power-law* (with an order of $1/7$) inflow profiles of the bare rod bundle are displayed in Fig. D.3.

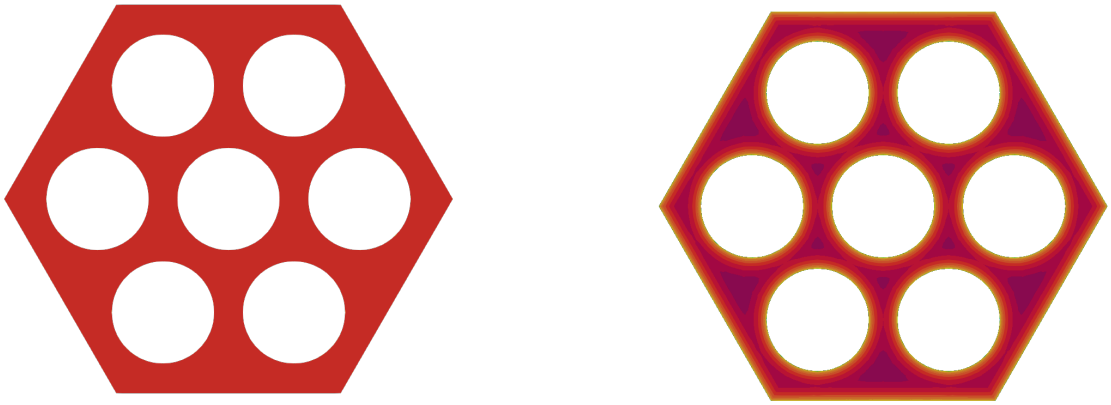


Figure D.3: Uniform (left) and power-law (right) inflow profiles of the seven-pin bare rod bundle.

The parameters of rod bundle cases are sampled in two steps. Firstly, 19 uniformly distributed grid points within the ranges are chosen (Fig. D.4). Then, Algorithm 4 is applied to add other parameters to reach the total amount N_μ.

After the two steps, the final $N_\mu = 50$ samples are selected (see Fig. D.5).

To identify the effects of the number of training points, four snapshot sizes are applied to the constructed ROMs. They consist of 5, 10, 20, and 30 points. The training sets overlap due to the greedy algorithm, see details in Algorithm 3. See Fig. D.6 for more information.

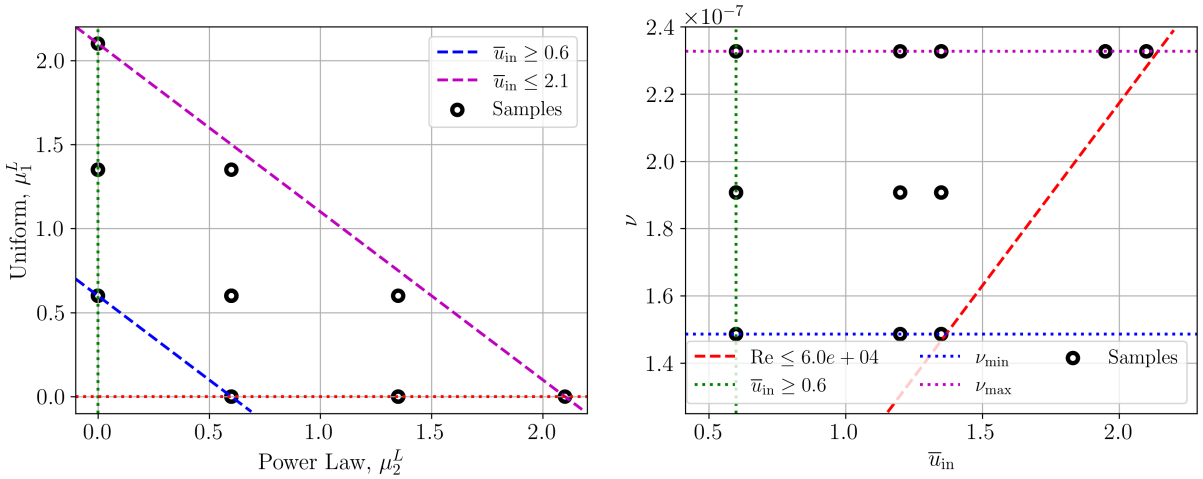


Figure D.4: Pre-selected uniformly distributed grid points within the ranges. The values are constraints to ensure the limits, i.e., $Re \in [1.4, 6.0] \times 10^4$ and $\overline{u}_{\text{in}} \in [0.6, 2.1]$.

Algorithm 4 Expansion of P with elements from a candidate set P_c

1: **Input:** $P = \{\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_{\tilde{N}_\mu}\}$, each $\boldsymbol{\mu}_i = [\nu_i, \mu_{1,i}^L, \mu_{2,i}^L]^{\mathrm{T}}$, candidate set P_c, number of new points N_{new}
2: **Output:** New P containing added parameters, total number of parameters N_μ
3: Compute the full centroid:

$$\overline{\boldsymbol{\mu}} = \frac{1}{N_\mu} \sum_{\boldsymbol{\mu} \in P} \boldsymbol{\mu}$$

4: **for** $k = 1$ to N_{new} **do**
5: Compute the centroid of current P:

$$\overline{\boldsymbol{\mu}} = \frac{1}{|P|} \sum_{\boldsymbol{\mu} \in P} \boldsymbol{\mu}$$

6: Select:

$$\boldsymbol{\mu}^{(k)} = \arg\max_{\boldsymbol{\mu} \in P_c} \|\boldsymbol{\mu} - \overline{\boldsymbol{\mu}}\|_{L^2}$$

7: Update $P \leftarrow P \cup \{\boldsymbol{\mu}^{(k)}\}$, $P_c \leftarrow P_C \setminus \{\boldsymbol{\mu}^{(k)}\}$
8: **end for**

D.2.2 PODI models

The comparisons of several kernels of POD-RBF and POD-GPR approaches are illustrated in Figs. D.7 and D.8. The lines are obtained with $N_{\text{train}} = 5, 10, 20,$ and $, 30$, and validated with a standard test set $N_{\text{test}} = 20$.

For RBF, the *Cubic* is suitable for $N_{\text{train}} = 5$ and 10. In contrast, for larger sets, *Quintic* is better. For GPR, *Matern* are the best except for $N_{\text{train}} = 5$.

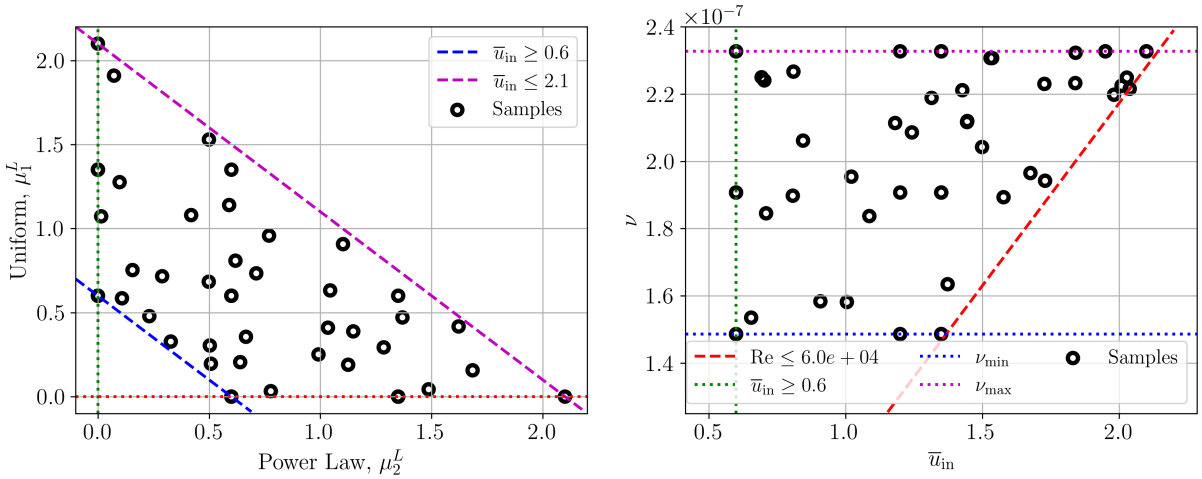


Figure D.5: Samples for the bare rod bundle case containing uniform grid points and randomly generated values. The values are chosen to ensure the limits, i.e., $Re \in [1.4, 6.0] \times 10^4$ and $\overline{u}_{\text{in}} \in [0.6, 2.1]$.

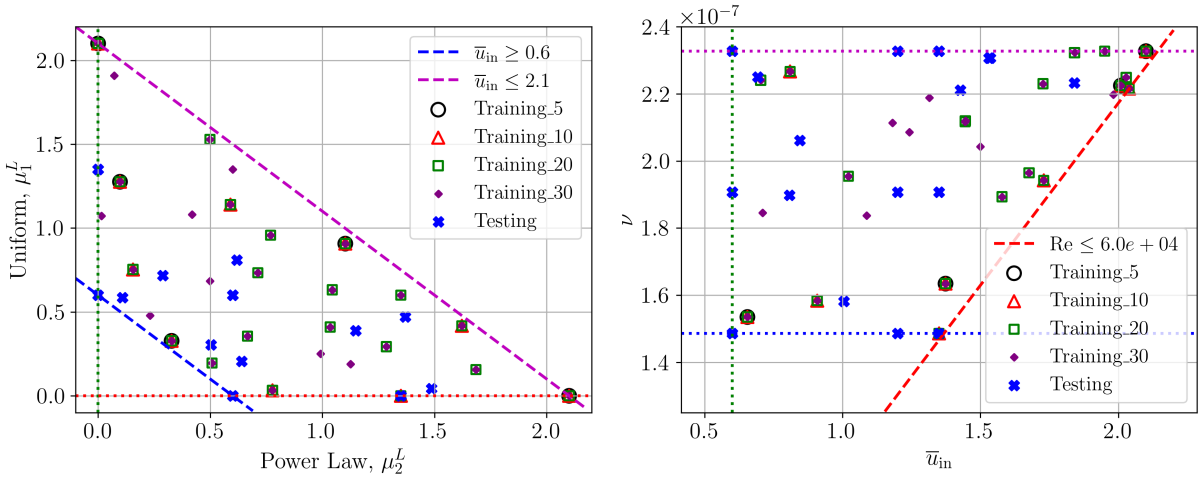
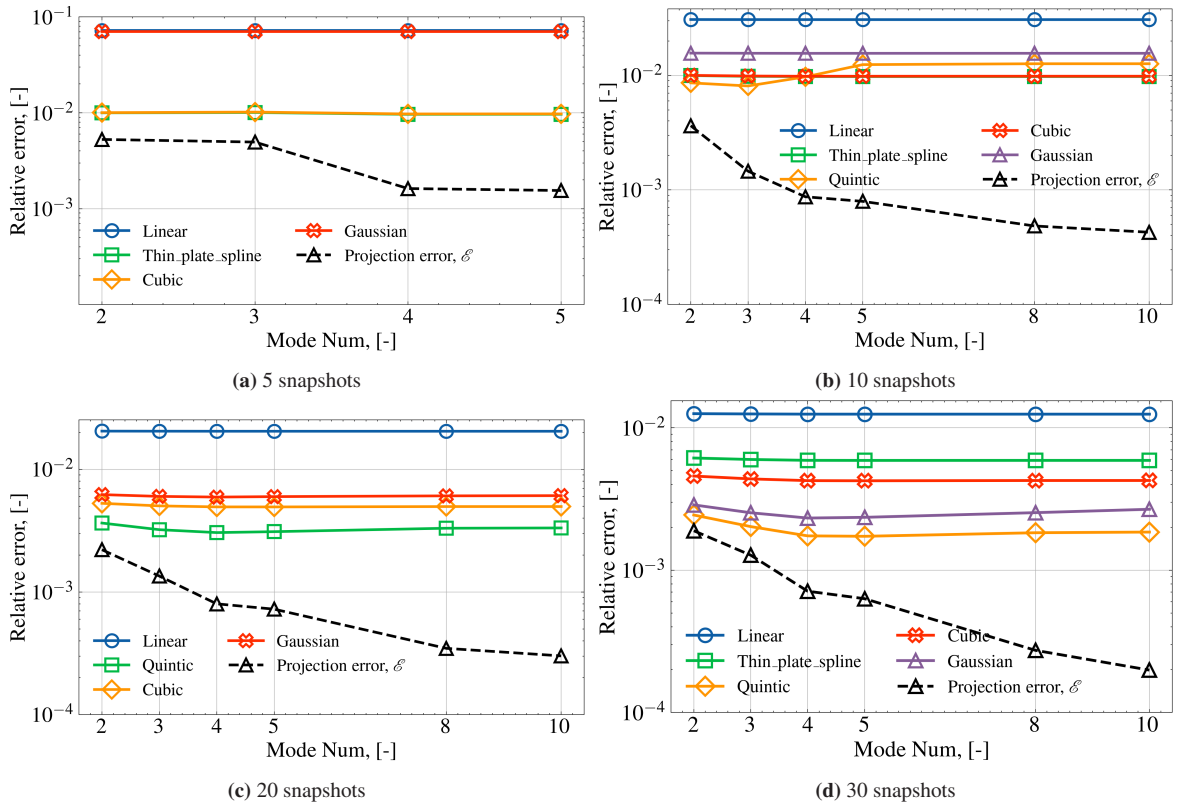


Figure D.6: Four sizes of the training and test sets. They are 5, 10, 20, and 30 points for constructing ROMs for bare-rod bundle cases. Due to the use of the greedy algorithm, the training sets overlap.

Fig. D.9 shows the influence of the architecture of POD-ANN ROMs. The settings are listed in Table 4.3. The effects of activation functions are presented in Fig. D.9. Consequently, a network structure of 3 hidden layers and 10 neurons per layer, and *Tanh* are selected for the simulations.

For the bare rod bundle case, the effects of activation functions of POD-ANN ROMs are also studied (see Fig. D.10). The results are obtained with $N_{\text{train}} = 30$ and $N_{\text{test}} = 20$. The velocity and eddy viscosity modes range from 2 to 10, and the pressure mode is set to 2. Based on the comparisons, *Tanh* is selected for building surrogates as follows.

The computational costs of FOM and ROMs are summarized in Table D.2. The speedup ratio is around 10^4 when only the solving time is considered. The quantities are more than 10^2, even including the POD and ROM construction (i.e., projection for POD-Galerkin and training for PODI). It is clear that *Eigen* [387] is faster than *NumPy* [369] for calculating the POD modes.



(a) 5 snapshots

(b) 10 snapshots

(c) 20 snapshots

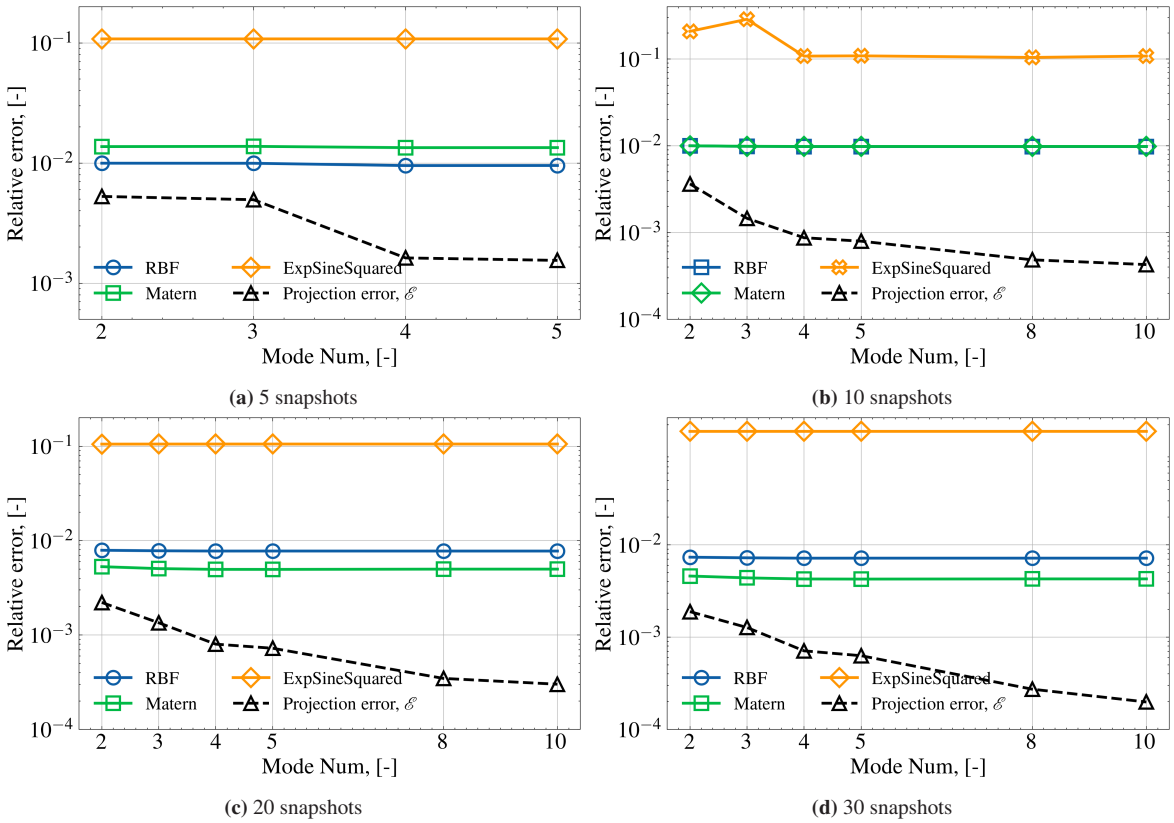
(d) 30 snapshots

Figure D.7: Effects of kernels of POD-RBF ROMs for the bare rod bundle case with different sizes of training set, indeed, $N_{\text{train}} = 5, 10, 20, 30$, and $N_{\text{test}} = 20$.

Be aware that the PODG-Galerkin ROM can simultaneously predict velocity, pressure, and eddy viscosity, whereas PODI ROMs require building three separate models. Also, the PODI costs for the three variables differ. Velocity has three components, while pressure and eddy viscosity are scalars. Thus, the cost of calculating POD modes is about one-third that of calculating velocity. Nevertheless, the duration of *Interpolation* and *Solving* for the three surrogates is similar.

Table D.2: Computational time of FOM and ROM calculations with $N_{\text{train}} = 30$, running on an Intel Xeon Platinum 8358 CPU (64 cores, 2.6 GHz). The FOM time is per simulation. The ROM time is the sum of the durations for 20 test samples. The POD-Galerkin ROM employs lifting functions and the supremizer. The PODI-RBF uses the *Quintic* kernel, the PODI-GPR adopts *Matern* kernel, and the PODI-ANN has three hidden layers and 10 neurons per layer with *Tanh* activation function.

	Item	FOM	Galerkin	RBF	GPR	ANN
Offline	FOM (per)	2150 s		-		
	POD	-	7.1 s		9.5 s	
	Supremizer	-	18.1 s		-	
	Projection	-	19.8 s		-	
Online	Interpolation	-	-	1.0e-3 s	1.5e-3 s	17.0 s
	Solving ($N_{\text{test}} = 20$)	-	1.0e-2 s	1.0e-3 s	1.0e-3 s	0.1 s



(a) 5 snapshots

(b) 10 snapshots

(c) 20 snapshots

(d) 30 snapshots

Figure D.8: Effects of kernels of POD-GPR ROMs for the bare rod bundle case with different sizes of training set, indeed, $N_{\text{train}} = 5, 10, 20, 30$, and $N_{\text{test}} = 20$.

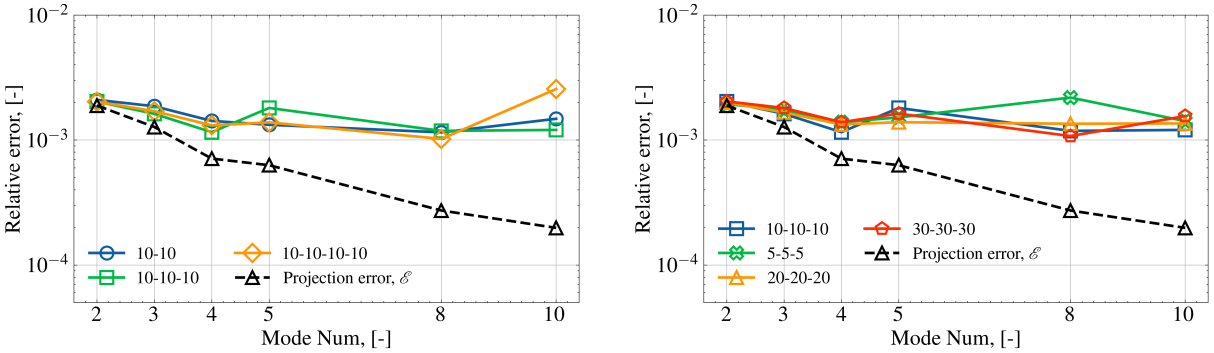


Figure D.9: Effects of number of hidden layers (left) and neurons per layer (right) of POD-ANN ROMs for the bare rod bundle case. The results are obtained with $N_{\text{train}} = 30$, and validated with a standard test set $N_{\text{test}} = 20$.

D.3 A wire-wrapped bundle

The two inflow conditions are illustrated in Fig. D.11.

The pre-selected grid points are plotted in Figs. D.12. By employing the algorithm 4, the final $N_\mu = 50$ samples are selected (see Fig. D.13).

To identify the effects of the number of training points, another four sets with 5, 10, 20, and 30 snapshots are also considered. They are shown in Fig. D.14.

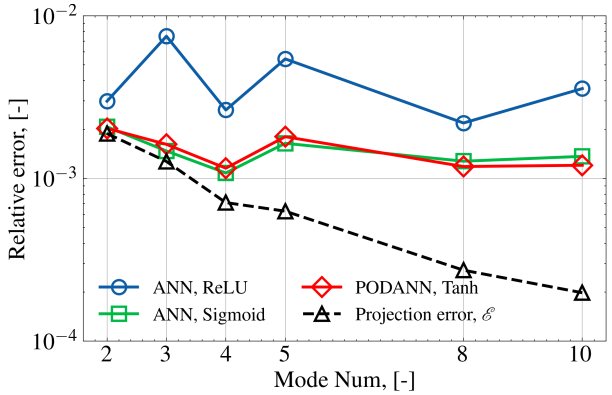


Figure D.10: Effects of activation functions of POD-ANN ROMs for the bare rod bundle case, with $N_{\text{train}} = 30$ and $N_{\text{test}} = 20$.

Figure D.11: Uniform (left) and power-law (right) inflow profiles of the seven-pin wire-wrapped rod bundle.

The comparisons of several kernels of POD-RBF and POD-GPR approaches are illustrated in Figs. D.15 and D.16. The lines are obtained with $N_{\text{train}} = 5, 10, 20,$ and $, 30$, and validated with a standard test set $N_{\text{test}} = 20$.

To determine the architecture of POD-ANN ROMs, the effects of the number of hidden layers and the number of neurons per layer are studied. Finally, the same architecture as in previous cases, which contains three hidden layers with 10 neurons per layer and *Tanh*, is selected for the regression.

The computational costs of FOM and ROMs are summarized in Table D.3. It can be seen that a single FOM simulation takes around 18.57 hours. In contrast, the time for approximations using POD-Galerkin and PODI ROMs for 20 test samples is only 0.05 seconds and around 0.001 seconds, respectively. The speedup ratio is around 10^6 when only the online solving time is considered. In addition, for these complex models, *Eigen* library [387] is much faster than *NumPy* package [369] for performing the POD.

One can see that the time required for interpolation and solving PODI models is similar to that for the bare rod bundle case. This is because the algorithms are designed to estimate surrogate parameter-to-mode coefficients, which are independent of the FOM size. The PODI times for the three variables are also different for the same reason as in the bare rod bundle case.

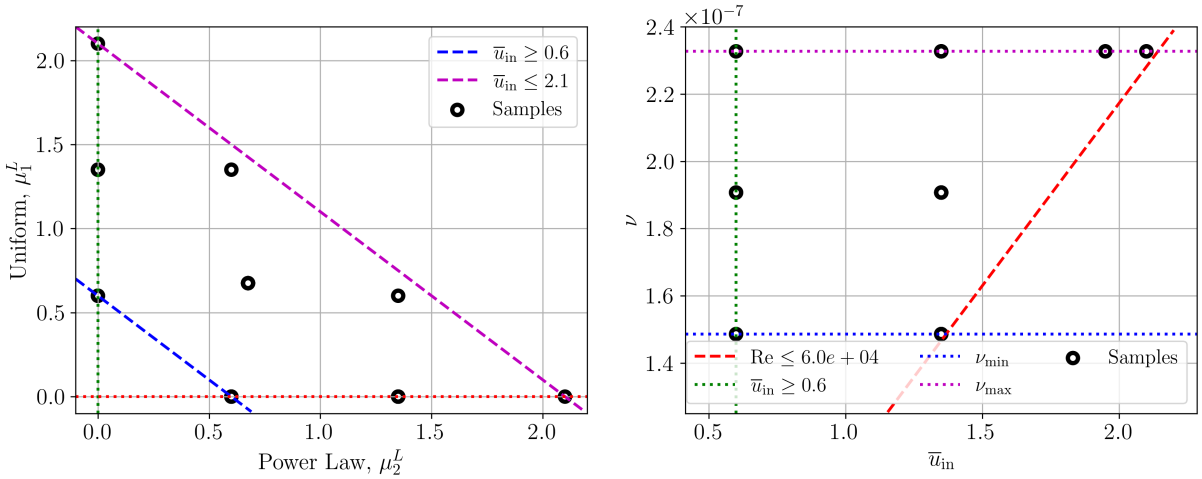


Figure D.12: Pre-selected uniformly distributed grid points within the ranges.

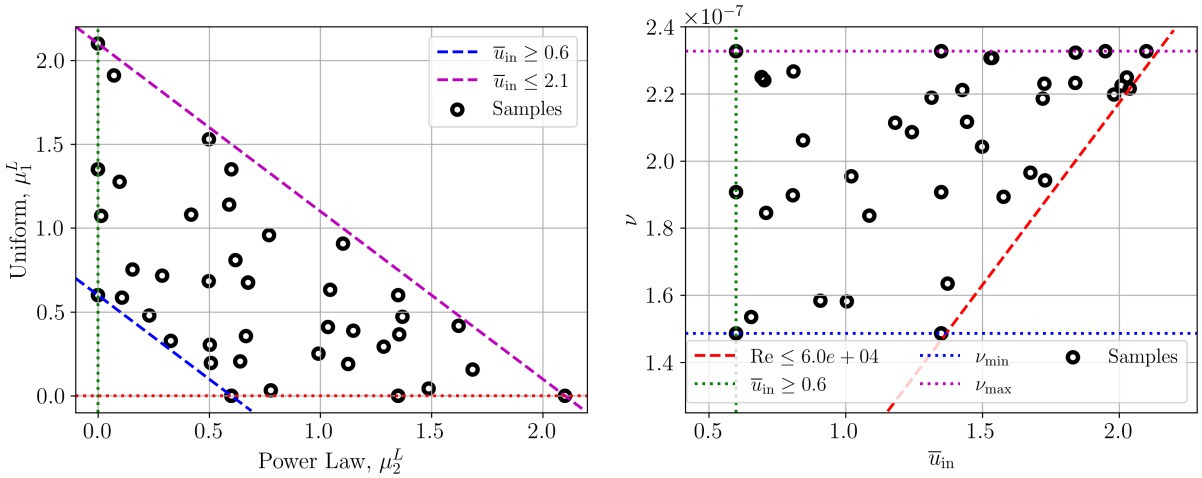


Figure D.13: Fifty samples for the wire-wrapped rod bundle case containing uniform grid points and randomly scattered values. The values are chosen to ensure the limits, i.e., $Re \in [1.4, 6.0] \times 10^4$ and $\overline{u}_{\text{in}} \in [0.6, 2.1]$.

Note that not all file input and output (I/O) operations are included in the reported computational times, which are not negligible for large-scale problems. However, since the focus of this dissertation is on the development of ROMs, the detailed analysis of each operation is not presented. Actually, the total time of ROM is relatively short, so it will be affected by many factors, e.g., the hardware and the storage device. Thus, the speedup ratios are just for reference.

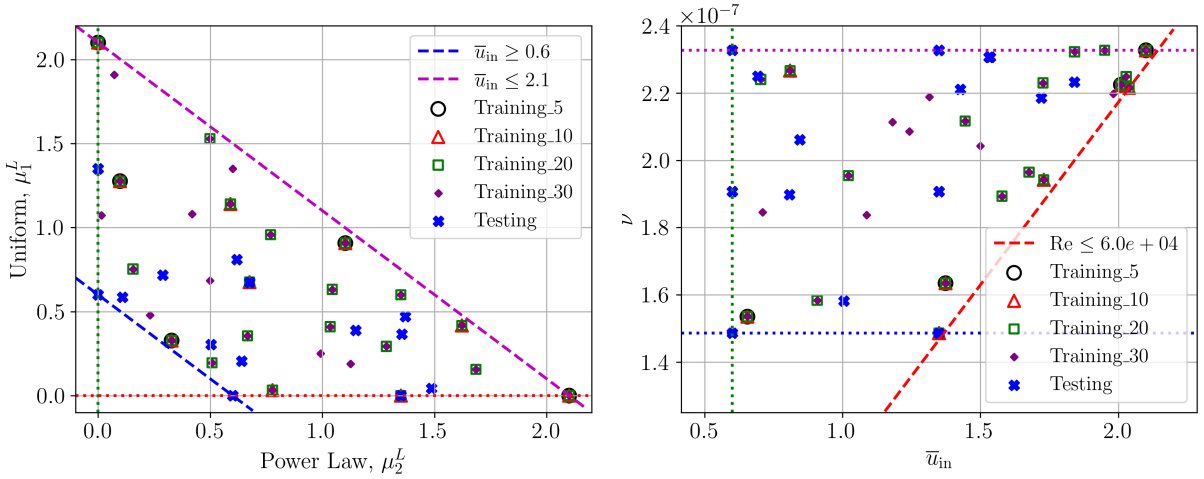


Figure D.14: Four sizes of the training sets and the test set. They are 5, 10, 20, and 30 points for constructing ROMs for wire-wrapped bundle cases.

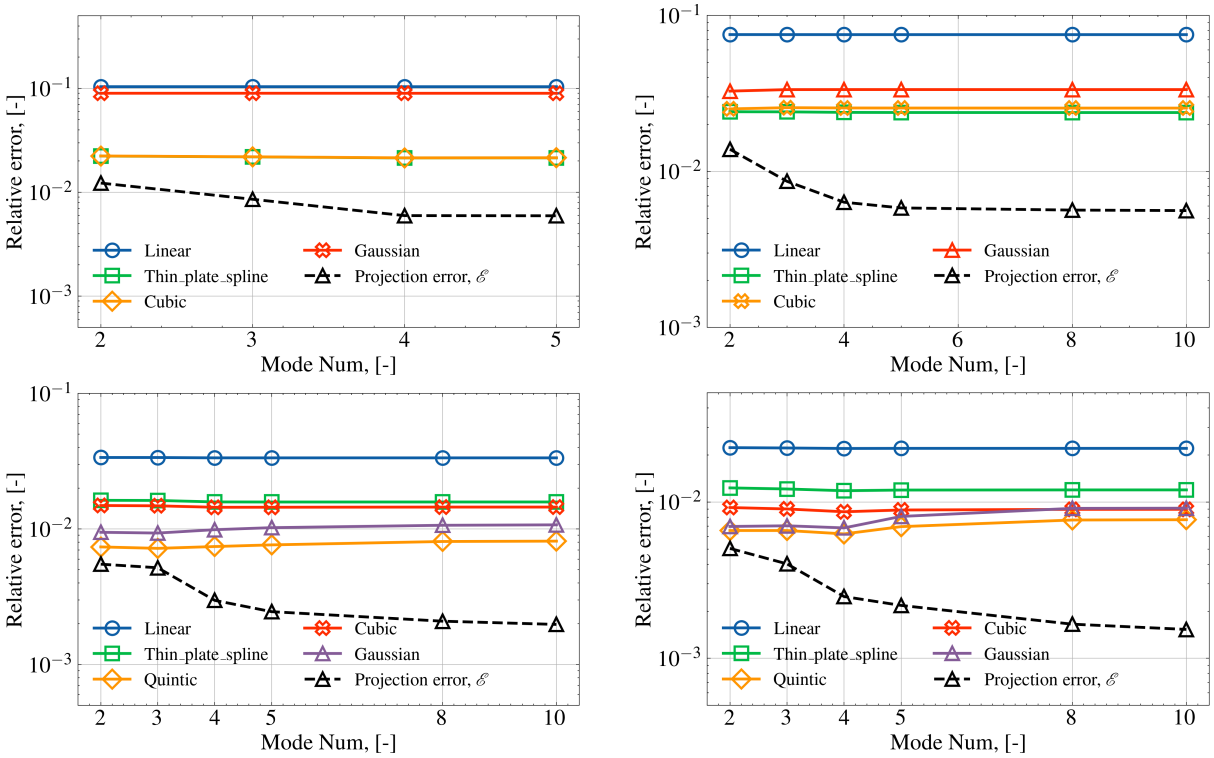


Figure D.15: Effects of kernels of POD-RBF ROMs for the wire-wrapped rod bundle case with different sizes of training set, indeed, $N_{\text{train}} = 5, 10, 20, 30$, and $N_{\text{test}} = 20$.

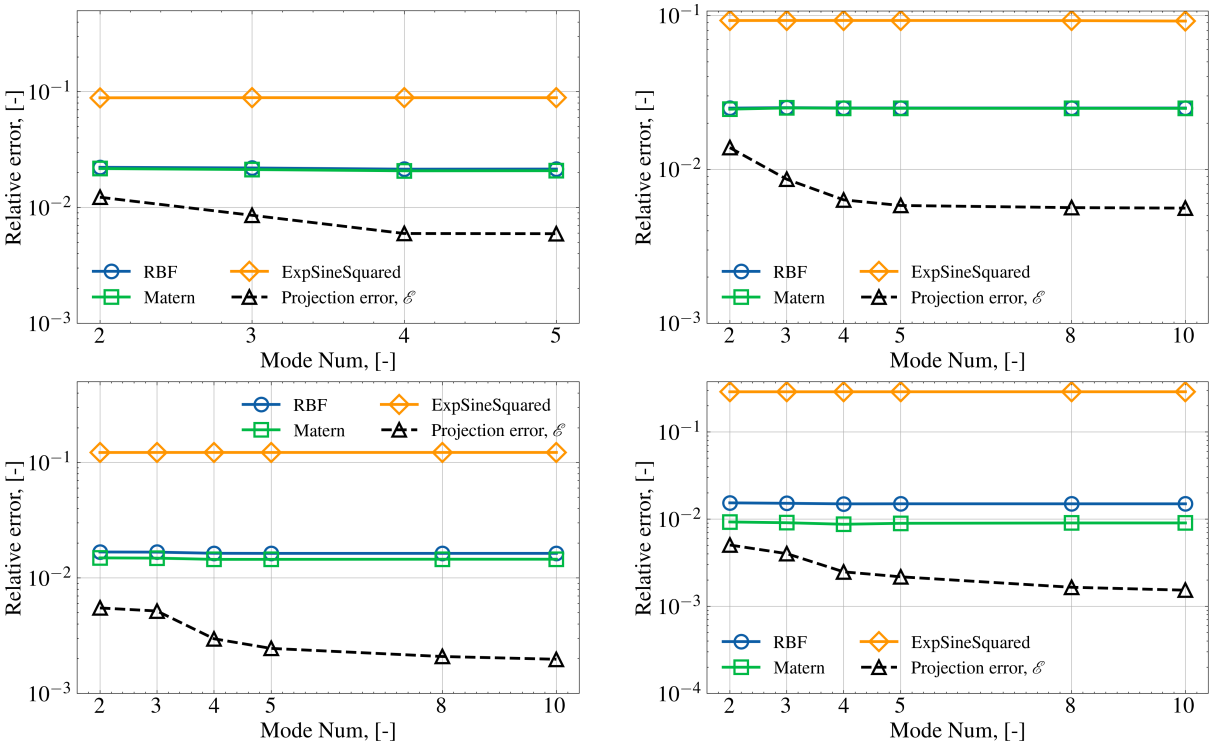


Figure D.16: Effects of kernels of POD-GPR ROMs for the wire-wrapped rod bundle case with different sizes of training set, indeed, $N_{\text{train}} = 5, 10, 20, 30$, and $N_{\text{test}} = 20$.

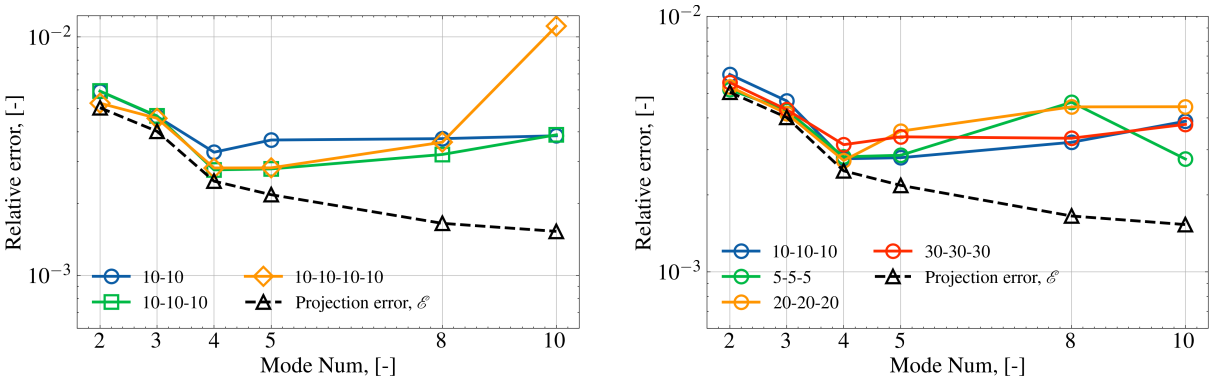


Figure D.17: Comparisons for the number of hidden layers (left) and neurons per layer (right) of POD-ANN ROMs for the wire-wrapped rod bundle case. The results are obtained with $N_{\text{train}} = 30$, and validated with a standard test set $N_{\text{test}} = 20$.

Table D.3: Computational time of FOM and ROM calculations with $N_{\text{train}} = 30$, running on an Intel Xeon Platinum 8358 CPU (64 cores, 2.6 GHz). The FOM time is per simulation. The ROM time is the sum of the durations for 20 test samples. "POD" refers to the Proper Orthogonal Decomposition step, which is common to all ROMs. The "Galerkin" column refers to the intrusive POD-Galerkin ROM. In contrast, the "RBF", "GPR", and "ANN" columns correspond to non-intrusive PODI (POD with Interpolation) ROMs using Radial Basis Function, Gaussian Process Regression, and Artificial Neural Network surrogates, respectively. The POD-Galerkin ROM employs lifting functions and the supremizer. The PODI-RBF uses the *Quintic* kernel, the PODI-GPR adopts the *Matern* kernel, and the PODI-ANN has three hidden layers and 10 neurons per layer with *Tanh* activation function.

	Item	FOM	Galerkin	RBF	GPR	ANN
Offline	FOM (per)	18.57 h				
	POD	-	33.5 s		116.7 s	
	Supremizer	-	112.9 s		-	
	Projection	-	243.5 s		-	
Online	Interpolation	-	-	1.0e-3 s	1.5e-3 s	17.1 s
	Solving ($N_{\text{test}} = 20$)	-	5.0e-2 s	1.0e-3 s	1.5e-3 s	0.1 s

Acknowledgement

To all who have supported me, trusted me, believed in me, encouraged me, inspired me, accompanied me, challenged me, hurt me, and taught me along the way.

To anything that I have experienced, enjoyed, and suffered through in the past few years.

To every piece of land I have stepped on, passed by, or even just glanced at.

No matter how it starts and ends, the journey itself is always worthwhile. I am enjoying every moment.

I know I will arrive here, one day, on one occasion. But finally, I am here, at this moment.

It is a long journey of four years. I have experienced a lot, good and bad, happy and sad, exciting and boring, hopeful and frustrating. Every interaction with the world, i.e., the environment and people, shaped me to be who I am today. With kindness, patience, encouragement, and support from so many people, I have finally reached this point.

I should say thanks to every person who has appeared in my life during these years. We have influenced each other, more or less, deeply or slightly. Sincerely, I want to list all the kindness I have received, but unfortunately, time fades my memory, since the past and from now onward. Pitifully, my biological brain is not as persistent as digital storage. I appreciate every single moment throughout these years.

I want to send my deepest gratitude to my *Doktorvater*, Prof. Dr.-Ing. (habil.) Andreas G. Class. I learned from him about academic knowledge and the attitude towards research and life. His patience and encouragement are always online to me. Our interactions include many life aspects, which actually released me from stress and pressure during the PhD journey. It is frustrating for me that I missed our first in-person meeting because I was not on time for the bus. But you don't say anything about it, and just welcome me warmly. The first half year of my stay here was the COVID-19 time. Few colleagues are in the office. We have a chance to talk a lot during lunch breaks. In fact, my English improved dramatically during these talks, which is significant for my integration and confidence. You also shared your life experience with me as examples, which are inspiring and enlightening. Many things happened during the years. Thank you, Andreas, for your continued support and understanding.

Appreciate the help from Prof. Dr.-Ing. Cheng, Xu. Without your recommendation, I would not have had the chance to join the ITES and start my PhD journey in Germany. Your support gives me the opportunity to explore the world beyond my previous experience. Besides, I will sometimes recall your words to encourage our Chinese students to help each other and build a community here in Karlsruhe. Your kindness and generosity impressed me a lot. Social responsibility is essential for every individual, especially for those who have the ability to help others. Thank you, Prof. Cheng.

The stay in SISSA is a valuable experience for me to enhance my understanding of reduced order modeling and data-driven methods. I want to thank my host, Prof. Gianluigi Rozza, for providing me with this opportunity. Your group is so active and energetic. I learned a lot from the group meetings and discussions with your students and postdocs. You are always busy, but you still find time to help me with my questions. Thank you, Prof. Rozza, for your guidance to help me extend my research horizon in the field of reduced order modeling.

Thanks to the supervisor of my master's degree, Prof. Wen, Qinglong. You helped to build the contact and make my trip come true. Your encouragement helped me to move forward.

It is a lucky experience that all colleagues in the institute are so kind and friendly. Although I am the only guy who can not speak German, you decided to use English for group meetings and lunch time, so that I can join the discussions.

Thanks to my Palestinian "father", Dr. Abdalla Batta. I will always remember your word: "Don't worry, my son. You can always ask me for help at any time if you want." You guided me to use my intelligence to think and solve engineering problems. Hope peace will come to the world. Hope you have the chance to go back to Gaza and enjoy the days there. Hope you healthy and take care of yourself. Thank you, Abdalla.

Dr. Karsten Litfin is the first colleague to show me around the institute. I still remember your warm welcome on my first day here. You show me around the buildings, and especially my office, where I spend most of my time. It is also interesting to see your DIY creations. Thank you, Karsten, for your help in my first step.

Dr. Yáñez Escanciano, Jorge, is the guy who provided me the most help in writing, not just scientific writing, but also general writing skills. The days working with you on revising my papers are unforgettable. We went through every single sentence again and again. You taught me how to express ideas clearly, logically, concisely, and understandably. Thank you, Jorge, for your patient guidance and support. Meanwhile, our talks on communism and capitalism are also very interesting. I always believe the basic virtues of human beings, sincerity, kindness, and mutual respect, understanding and support, are the foundation of socialism and communism. I appreciate you that I have noticed those merits from you actions and talks.

Also, the best wishes to our ITES "nerds". Dr. Philipp Marthaler, Dr. Lorenz Weber, Nicolas Wefers, and Lyu, Biao. It is impressive that you are all talented and hard-working guys. I have to say I am jealous of your intelligence. Meanwhile, your hard-working spirit is really inspiring.

That something shocks me, as the saying goes, "The desperate fact is that the smarter guy works harder than you."

I am happy to share the office with Philipp, unless he became "annoyed" after finishing his dissertation and later his defense, haha. I will never forget that you visited me during my stay in Trieste. Our car trip to Croatia is so relaxing and enjoyable. Thank you, Philipp. Note that your name is still on the office door after you left and witnessed my whole PhD journey here. Wish you all the best and be a big boss in the future as you are talented and capable in both technical and management aspects.

Congratulations to Lorenz for his successful defense. Your presentation is impressive. As I said, it is the best presentation I have ever seen. Every slide is clear, and your explanation is easy to follow. No matter whether I left the office at 7 am or 8 am, your office light is always on. Thank you, the crazy and creative Lorenz. I believe you will make noticeable achievements in academic. In the meantime, enjoy sailing and hiking.

Nicolas is another hard-working and talented doctoral student in ITES. The discussion with you is always impressive. You really managed to present your idea in depth and guided our talks beyond the surface. I still remember some of our chats in Andreasbräu. Thank you, Nicolas, you are a warm-hearted and smart guy. Hope you a successful work, but do not forget to rest more and have fun along the way.

Biao, my Chinese colleague in ITES. Your broad knowledge of many aspects impressed me a lot. We had many discussions during lunch breaks about politics, history, culture, finance, sports, health, and so on. It is amazing that you know deeply about those topics and you share your own insights. Your sharing is a window for me to learn more about Germany. Thank you, Biao, for your friendship and support.

I would also like to thank Prof. Leo Bühler, Prof. Daniel Banuti, and Dr. Ivan Otic. The interactions with you are always inspiring. Additionally, the support from the administrative staff is necessary for my research. Thanks to Ms. Heide Hofmann, Ms. Friederike Bonsack, Mr. Eugen Urbach, and Mr. Ronni Günther.

In 2024, I visited the mathLab, Scuola Internazionale Superiore di Studi Avanzati (SISSA) in Trieste, Italy, for six months. The visit is supported by KHYS (Karlsruhe House of Young Scientists) under the program of "Research Travel Grant". I appreciate KHYS for the support.

During my stay in SISSA, I worked with Guglielmo Padula. Guglielmo, you are one of the best programmers I have ever met. Your coding skills are impressive. Dr. Ivan Prusak helped me a lot in understanding some mathematical concepts corresponding to the ROM. Besides, I want to thank Prof. Giovanni Stabile, Anna Ivagnes, Pierfrancesco Siena, and all other people in the mathLab group. Thank you all for your help and support during my stay in Trieste.

Many supports come from my friends, in Karlsruhe, in Trieste, in Germany, in Europe, and in China. Too many names should be mentioned here, but I have to fold them into a general "thank

you". A lot of guys have continuously supported me, some of them I have known for more than ten years, and some of them I met in Karlsruhe and on my visit to SISSA. Thank you all, my dear friends. Your friendship is precious to me.

My doctoral research is supported by the China Scholarship Council (CSC) and ITES, KIT. I want to thank CSC for the financial support during my PhD journey. Also, I want to thank ITES for providing me with a good research environment and the necessary resources.

The simulations shown in this dissertation were carried out on the BwUniCluster provided by the SCC at KIT. I want to thank the SCC staff for their support in using the high-performance computing resources. I also want to appreciate the two open-source software communities, OpenFOAM (`https://develop.openfoam.com/Development/openfoam.git`) and ITHACA-FV (`https://github.com/ITHACA-FV/ITHACA-FV.git`).

Finally, I believe I can not achieve this milestone without the support of my family. Therefore, my last and most important acknowledgement goes to my family. My mother and father are working extremely hard to take me from a poor village in China to here. They sacrificed so much for my education and life. Mom and Dad, your unconditional love and support are the backbone of my life. No matter how far I am away from you, I always feel your care and love.

搁笔之际，心怀怅然。
谨录稼轩词意，赠故友亲朋，并寄共度之韶华。
纸短情长，或可慰怀。

鹧鸪天·送人

辛弃疾〔宋代〕

唱彻阳关泪未干，功名馀事且加餐。
浮天水送无穷树，带雨云埋一半山。

今古恨，几千般，只应离合是悲欢?
江头未是风波恶，别有人间行路难!

List of Publications

Journal articles

[1] Shenhui Ruan, Jorge Yanez, and Andreas G. Class. Local reduced subspaces of subchannel-inspired subdomains. *International Journal for Numerical Methods in Engineering*, 125 (18):e7552, 2024.

[2] Shenhui Ruan, Andreas G. Class, and Gianluigi Rozza. Introduction to the combination of reduced order models and domain decomposition: State of the art and perspectives, January 2026.

[3] Shenhui Ruan, Ivan Otic, and Andreas G. Class. Simplified modeling of a wire-wrapped bundle employing a momentum source methodology. 2026.

Conference contributions

[1] Shenhui Ruan, Jorge Yanez, and Andreas G. Class. Distances between proper orthogonal decomposition reduced subspaces of repeating subdomains. The 9th European Congress on Computational Methods in Applied Sciences and Engineering (ECCOMAS 2024), 06 2024.

[2] Shenhui Ruan, Andreas G. Class, Giovanni Stabile, and Gianluigi Rozza. Domain decomposition reduced order model for large scale industrial facilities consisting of repeating subdomains. 11th edition of the International Conference on Computational Methods for Coupled Problems in Science and Engineering (COUPLED 2025), 05 2025.

Open source codes

[1] Shenhui Ruan. foamToPython: An IO package for OpenFOAM and Numpy, 10 2025. URL `https://github.com/Ruansh233/foamToPython.git`.

[2] Shenhui Ruan. PODImodels: Model Order Reduction using Proper Orthogonal Decomposition with Interpolation, 09 2025. URL `https://github.com/Ruansh233/PODImodels.git`.

Bibliography

[1] NASA Goddard Institute for Space Studies. Global land-ocean temperature index, 2024. URL https://climate.nasa.gov/vital-signs/global-temperature/?intent=121.

[2] NOAA National Oceanic & Atmospheric Administration. Monthly average mauna Loa CO_2, 2024. URL https://gml.noaa.gov/ccgg/trends/.

[3] NASA. What is climate change? - NASA science. Technical report, NASA, 2024. URL https://science.nasa.gov/climate-change/what-is-climate-change/.

[4] United Nations. Renewable energy - powering a safer future, 2023. URL https://www.un.org/en/climatechange/raising-ambition/renewable-energy.

[5] Katy Laffan. International day of clean energy: Why nuclear power? Technical report, International Atomic Energy Agency, 1 2024. URL https://www.iaea.org/newscenter/news/international-day-of-clean-energy-why-nuclear-power.

[6] EPA. Atmospheric concentrations of greenhouse gases. Technical report, United States Environmental Protection Agency, 2022.

[7] Wikipedia-contributors. Greenhouse gas, 2 2002. URL https://en.wikipedia.org/wiki/Greenhouse_gas#cite_note-5.

[8] IEA. Renewables 2023. Technical report, International Energy Agency, 2024. URL https://www.iea.org/reports/renewables-2023.

[9] DOE and GIF. A technology roadmap for generation IV nuclear energy systems. Technical report, U.S. DOE Nuclear Energy Research Advisory Committee and Generation IV International Forum, 2002.

[10] GIF. Technology roadmap update for generation IV nuclear energy systems. Technical report, Generation IV International Forum, 2014.

[11] Yu Liang, Dalin Zhang, Jing Zhang, Xiao Liu, Yapeng Liu, Lei Zhou, Yutong Chen, Xiaoyu Wang, Wenxi Tian, Suizheng Qiu, et al. A subchannel analysis code SACOS-Na for sodium-cooled fast reactor. *Progress in Nuclear Energy*, 166:104959, 2023.

[12] Jaehyuk Eoh, Chang-Gyu Park, Jae-Yong Lim, June-Hyung Kim, Huee-Youl Ye, and Jinwook Chang. Design and safety features of SALUS-100: A long fuel-cycled sodium-cooled fast reactor. *Nuclear Engineering and Design*, 420:112996, 2024.

[13] Neil E Todreas and Mujid S Kazimi. *Nuclear Systems Volume I: Thermal Hydraulic Fundamentals*. CRC press, 2021.

[14] Neil E Todreas, Mujid S Kazimi, and Mahmoud Massoud. *Nuclear Systems Volume II: Elements of Thermal Hydraulic Design*. CRC Press, 2021.

[15] Ferry Roelofs. *Thermal Hydraulics Aspects of Liquid Metal Cooled Nuclear Reactors*. Woodhead Publishing, 2018.

[16] A Moorthi, Anil Kumar Sharma, and K Velusamy. A review of sub-channel thermal hydraulic codes for nuclear reactor core and future directions. *Nuclear Engineering and Design*, 332:329–344, 2018.

[17] Jaewoon Yoo, Jinwook Chang, Jae-Yong Lim, Jin-Sik Cheon, Tae-Ho Lee, Sung Kyun Kim, Kwi Lim Lee, and Hyung-Kook Joo. Overall system description and safety characteristics of prototype gen IV sodium cooled fast reactor in Korea. *Nuclear Engineering and Technology*, 48(5):1059–1070, 2016.

[18] Chong Zhou, Klaus Huber, and Xu Cheng. Validation of the modified ATHLET code with the natural convection test of the PHENIX reactor. *Annals of Nuclear Energy*, 59:31–46, 2013.

[19] RL Sun, DL Zhang, Y Liang, MJ Wang, WX Tian, SZ Qiu, and GH Su. Development of a subchannel analysis code for SFR wire-wrapped fuel assemblies. *Progress in Nuclear Energy*, 104:327–341, 2018.

[20] Chaehyuk Im, Kyongwon Seo, Jee-Hyun Cho, Ilhoon Jang, and Simon Song. Flow analysis in a 37-pin wire-wrapped rod bundle for sodium-cooled fast reactor using magnetic resonance velocimetry. I. validation and gross behavior. *Physics of Fluids*, 36(9), 2024.

[21] Guanwen Luo, Kuo Wang, Gang Tong, Chong Xie, Xiaohang Wu, Huiyong Zhang, Yiliang Xie, and Jiayi Chen. Thermal-hydraulic analysis of a 19-rod bundle LBE cooled fuel assembly with non-uniform rods power distribution by numerical simulation. *Annals of Nuclear Energy*, 215:111242, 2025.

[22] Stephen B Pope. *Turbulent flows*. Cambridge University Press, 2000.

[23] Henk Kaarle Versteeg. *An Introduction to Computational Fluid Dynamics the Finite Volume Method*. Pearson Education India, 2007.

[24] Fadl Moukalled, Luca Mangani, Marwan Darwish, F Moukalled, L Mangani, and M Darwish. *The Finite Volume Method*. Springer, 2016.

[25] Xiaoxi Zhang, Nan Gui, Hou-jun Gong, Xingtuan Yang, Jiyuan Tu, and Shengyao Jiang. Thermal hydraulic review of light water reactor based on subchannel code CTF. *Nuclear Engineering and Design*, 413:112482, 2023.

[26] Ferry Roelofs, VR Gopala, K Van Tichelen, X Cheng, E Merzari, and WD Pointer. Status and future challenges of CFD for liquid metal cooled reactors. *FR13, Paris, France*, 2013.

[27] Annette Heinzel, Wolfgang Hering, Jürgen Konys, Luca Marocco, Karsten Litfin, Georg Müller, Julio Pacio, Carsten Schroer, Robert Stieglitz, Leonid Stoppel, et al. Liquid metals as efficient high-temperature heat-transport fluids. *Energy Technology*, 5(7):1026–1036, 2017.

[28] Sara Perez-Martin, Thomas Schaub, Angela Jianu, Alexandru Onea, Wadim Jäger, Wolfgang Hering, and Robert Stieglitz. Research on liquid metal thermal-hydraulics at KIT. In *Meeting of the Expert Group on Liquid Metal Technology*, 2018.

[29] R Hu and TH Fanning. Progress report on development of intermediate fidelity full assembly analysis methods. Technical report, Argonne National Lab.(ANL), Argonne, IL (United States), 2011.

[30] Alfio Quarteroni, Andrea Manzoni, and Federico Negri. *Reduced basis methods for partial differential equations: an introduction*, volume 92. Springer, 2015.

[31] Gianluigi Rozza, Francesco Ballarin, Leonardo Scandurra, and Federico Pichi. *Real Time Reduced Order Computational Mechanics*. Springer, 2024.

[32] Bernard Haasdonk. Reduced basis methods for parametrized PDEs–a tutorial introduction for stationary and instationary problems. *Model reduction and approximation: theory and algorithms*, 15:65, 2017.

[33] Jan S Hesthaven, Gianluigi Rozza, Benjamin Stamm, et al. *Certified reduced basis methods for parametrized partial differential equations*, volume 590. Springer, 2016.

[34] Computational Fluid Dynamics Committee. Guide: Guide for the verification and validation of computational fluid dynamics simulations (AIAA G-077-1998 (2002)), 1998.

[35] William L Oberkampf and Timothy G Trucano. Verification and validation in computational fluid dynamics. *Progress in aerospace sciences*, 38(3):209–272, 2002.

[36] Iztok Tiselj, Cedric Flageul, and Jure Oder. Direct numerical simulation and wall-resolved large eddy simulation in nuclear thermal hydraulics. *Nuclear Technology*, 206(2):164–178, 2020.

[37] Daniele Dovizio, Blaž Mikuž, Afaque Shams, and Ferry Roelofs. Validating RANS to predict the flow behavior in wire-wrapped fuel assemblies. *Nuclear Engineering and Design*, 356:110376, 2020.

[38] Laura Iapichino, Alfio Quarteroni, and Gianluigi Rozza. Reduced basis method and domain decomposition for elliptic problems in networks and complex parametrized geometries. *Computers & Mathematics with Applications*, 71(1):408–430, 2016.

[39] Hengjie Wang, Robert Planas, Aparna Chandramowlishwaran, and Ramin Bostanabad. Mosaic flows: A transferable deep learning framework for solving PDEs on unseen domains. *Computer Methods in Applied Mechanics and Engineering*, 389:114424, 2022.

[40] Alessandro Petruzzi and Francesco D' Auria. Thermal-hydraulic system codes in nulcear reactor safety and qualification procedures. *Science and Technology of Nuclear Installations*, 2008(1):460795, 2008.

[41] Glenn A Roth and Fatih Aydogan. Theory and implementation of nuclear safety system codes–part II: System code closure relations, validation, and limitations. *Progress in Nuclear Energy*, 76:55–72, 2014.

[42] Khalil Moshkbar-Bakhshayesh and Soroush Mohtashami. Validation of codes for modeling and simulation of nuclear power plants: A review. *Nuclear Engineering and Design*, 421: 113120, 2024.

[43] Jian Deng, Qi Lu, Yu Liu, Xiaoyu Wang, Xiaolei Yang, Yong Zhang, and Jibin Zhang. Review of sub-channel code development for pressurized water reactor and introduction of CORTH-V2. 0 sub-channel code. *Progress in Nuclear Energy*, 125:103373, 2020.

[44] AS Mollah. Recent advances in computational convective heat transfer study in a sub-channel for nuclear power reactor and future directions. *WSEAS Transactions on Heat and Mass Transfer*, 18:162–181, 2023.

[45] Alfio Quarteroni and Silvia Quarteroni. *Numerical models for differential problems*, volume 2. Springer, 2009.

[46] Mingjun Wang, Yingjie Wang, Wenxi Tian, Suizheng Qiu, and GH Su. Recent progress of CFD applications in PWR thermal hydraulics study and future directions. *Annals of Nuclear Energy*, 150:107836, 2021.

[47] Ideen Sadrehaghighi. *Mesh Generation in CFD*. 01 2020.

[48] Claudio Canuto, M Youssuff Hussaini, Alfio Quarteroni, and Thomas A Zang. *Spectral methods*, volume 285. Springer, 2006.

[49] Alexandre Ern and Jean-Luc Guermond. *Theory and Practice of Finite Elements*, volume 159. Springer, 2004.

[50] Bernardo Cockburn, George E Karniadakis, and Chi-Wang Shu. *Discontinuous Galerkin Methods: Theory, Computation and Applications*, volume 11. Springer Science & Business Media, 2012.

[51] Brian Launder and Dudley Brian Spalding. *Mathematical Models of Turbulence*. Academic Press, Massachusetts, 1972.

[52] Suhas Vasant Patankar. *Heat and mass transfer in turbulent boundary layers*. PhD thesis, Imperial College London, 1967.

[53] Gosman AD, Pun WM, Akshai Kumar Runchal, Dudley Brian Spalding, and Wolfshtein M. Heat and mass transfer in recirculating flows. *Academic Press, London*, 1969.

[54] Akshai Kumar Runchal. *Transfer processes in steady two-dimensional separated flows*. PhD thesis, University of London, 1969.

[55] Elia Merzari, Aleks Obabko, Paul Fischer, Noah Halford, Justin Walker, Andrew Siegel, and Yiqi Yu. Large-scale large eddy simulation of nuclear reactor flows: Issues and perspectives. *Nuclear Engineering and Design*, 312:86–98, 2017.

[56] Jing Chen, Dalin Zhang, Ping Song, Xinan Wang, Shibao Wang, Yu Liang, Suizheng Qiu, Yapei Zhang, Mingjun Wang, and GH Su. CFD investigation on thermal-hydraulic behaviors of a wire-wrapped fuel subassembly for sodium-cooled fast reactor. *Annals of Nuclear Energy*, 113:256–269, 2018.

[57] F Roelofs, D Dovizio, H Uitslag-Doolaard, D De Santis, A Mathur, B Mikuz, and A Shams. Core thermal hydraulic CFD support for liquid metal reactors. *Nuclear Engineering and Design*, 355:110322, 2019.

[58] D Dovizio and B Mikuž. Pragmatic CFD approach for wire wrapped rod bundle. *SESAMED2*, 4:2703280–18, 2019.

[59] Xiang Chai, Lei Zhao, Wenjun Hu, Yun Yang, Xiaojing Liu, Jinbiao Xiong, and Xu Cheng. Numerical investigation of flow blockage accident in SFR fuel assembly. *Nuclear Engineering and Design*, 359:110437, 2020.

[60] Kefeng Lyu, Xudan Ma, Haitao Wang, Chong Chen, and Guangsheng Fei. CFD analysis of thermal-hydraluic behaviors in a LBE cooled 19-pin wire wrapped bundle under porous lumped blockage conditions. *Annals of Nuclear Energy*, 151:107956, 2021.

[61] Zhenhong Liu, Songbai Cheng, Wanai Li, and Wenhui Jin. Review of researches on flow blockage accident in liquid-metal-cooled reactor. *Progress in Nuclear Energy*, 176:105372, 2024.

[62] Brian Mays and R. Brian Jackson. Thermal hydraulic computational fluid dynamics simulations and experimental investigation of deformed fuel assemblies. 3 2017. doi: 10.2172/1346027. URL `https://www.osti.gov/biblio/1346027`.

[63] Andreas Wanninger. *Mechanical Analysis of the Bow Deformation of Fuel Assemblies in a Pressurized Water Reactor Core*. PhD thesis, Technische Universität München, 2018.

[64] YaoDi Li, Mei Huang, YanTing Cheng, Boxue Wang, and Xiangyuan Meng. Thermal hydraulic analysis of a deformed sfr fuel assembly using CFD. *Annals of nuclear energy*, 181:109520, 2023.

[65] Javier Martinez, Yu-Hsiang Lan, Elia Merzari, and Misun Min. On the use of LES-based turbulent thermal-stress models for rod bundle simulations. *International Journal of Heat and Mass Transfer*, 142:118399, 2019.

[66] E Merzari, WD Pointer, JG Smith, A Tentner, and P Fischer. Numerical simulation of the flow in wire-wrapped pin bundles: Effect of pin-wire contact modeling. *Nuclear engineering and design*, 253:374–386, 2012.

[67] Elia Merzari, Paul Fischer, H Yuan, Katrien Van Tichelen, Steven Keijers, Jeroen De Ridder, Joris Degroote, Jan Vierendeels, Heleen Doolaard, VR Gopala, et al. Benchmark exercise for fluid flow simulations in a liquid metal fast reactor fuel assembly. *Nuclear Engineering and Design*, 298:218–228, 2016.

[68] Landon Brockmeyer, Lane B Carasik, Elia Merzari, and Yassin Hassan. Numerical simulations for determination of minimum representative bundle size in wire wrapped tube bundles. *Nuclear Engineering and Design*, 322:577–590, 2017.

[69] Nolan Goth, P Jones, DT Nguyen, R Vaghetto, YA Hassan, A Obabko, E Merzari, and Paul F Fischer. Comparison of experimental and simulation results on interior subchannels of a 61-pin wire-wrapped hexagonal fuel bundle. *Nuclear Engineering and Design*, 338: 130–136, 2018.

[70] Elia Merzari, Aleksandr Obabko, Paul Fischer, and Manuele Aufiero. Wall resolved large eddy simulation of reactor core flows with the spectral element method. *Nuclear Engineering and Design*, 364:110657, 2020.

[71] E Baglietto, H Ninokata, and Takeharu Misawa. CFD and DNS methodologies development for fuel bundle simulations. *Nuclear Engineering and Design*, 236(14-16):1503–1510, 2006.

[72] D Angeli, A Fregni, and E Stalio. Direct numerical simulation of turbulent forced and mixed convection of LBE in a bundle of heated rods with P/D= 1.4. *Nuclear Engineering and Design*, 355:110320, 2019.

[73] Ferry Roelofs, Heleen Uitslag-Doolaard, Daniele Dovizio, Blaz Mikuz, Afaque Shams, Fulvio Bertocchi, Martin Rohde, Julio Pacio, Ivan Di Piazza, Graham Kennedy, et al. Towards validated prediction with RANS CFD of flow and heat transport in a wire-wrap fuel assembly. *Nuclear Engineering and Design*, 353:110273, 2019.

[74] A Shams, F Roelofs, E Baglietto, and EMJ Komen. High fidelity numerical simulations of an infinite wire-wrapped fuel assembly. *Nuclear Engineering and Design*, 335:441–459, 2018.

[75] Carolina Bourdot Dutra, Luiz Aldeia Machado, and Elia Merzari. Direct numerical simulation of heat transfer in a 7-pin wire-wrapped rod bundle. *Nuclear Science and Engineering*, 198(7):1439–1454, 2024.

[76] Gianluigi Rozza, Giovanni Stabile, and Francesco Ballarin. *Advanced Reduced Order Methods and Applications in Computational Fluid Dynamics*. SIAM, 2022.

[77] Roxana Crisovan, Davide Torlo, Rémi Abgrall, and S Tokareva. Model order reduction for parametrized nonlinear hyperbolic problems as an application to uncertainty quantification. *Journal of Computational and Applied Mathematics*, 348:466–489, 2019.

[78] Jorge Yanez and Andreas G Class. Analysis of the accuracy of residual heat removal and natural convection transients in reactor pools. *Nuclear Engineering and Design*, 378: 111151, 2021.

[79] Giuseppe Carere, Maria Strazzullo, Francesco Ballarin, Gianluigi Rozza, and Rob Stevenson. A weighted POD-reduction approach for parametrized PDE-constrained optimal control problems with random inputs and applications to environmental sciences. *Computers & Mathematics with Applications*, 102:261–276, 2021.

[80] Md Nurtaj Hossain and Debraj Ghosh. Adaptive reduced order modeling for nonlinear dynamical systems through a new a posteriori error estimator: Application to uncertainty quantification. *International Journal for Numerical Methods in Engineering*, 121(15): 3417–3441, 2020.

[81] Guglielmo Padula, Michele Girfoglio, and Gianluigi Rozza. A brief review of reduced order models using intrusive and non-intrusive techniques. *arXiv preprint arXiv:2406.00559*, 2024.

[82] Jian Yu, Chao Yan, and Mengwu Guo. Non-intrusive reduced-order modeling for fluid problems: A brief review. *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, 233(16):5896–5912, 2019.

[83] Steven K Thompson. *Sampling*, volume 755. John Wiley & Sons, 2012.

[84] Peter Benner, Serkan Gugercin, and Karen Willcox. A survey of projection-based model reduction methods for parametric dynamical systems. *SIAM review*, 57(4):483–531, 2015.

[85] T. Bui-Thanh, K. Willcox, and O. Ghattas. Model reduction for large-scale systems with high-dimensional parametric input space. *SIAM Journal on Scientific Computing*, 30(6): 3270–3288, January 2008. ISSN 1064-8275, 1095-7197. doi: 10.1137/070694855. URL `http://epubs.siam.org/doi/10.1137/070694855`.

[86] Michael D McKay, Richard J Beckman, and William J Conover. 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, 2000.

[87] Xirui Liu, Zhiyong Wang, Hongjun Ji, and Helin Gong. Application and comparison of several adaptive sampling algorithms in reduced order modeling. *Heliyon*, 10(15):e34928, August 2024. ISSN 24058440. doi: 10.1016/j.heliyon.2024.e34928. URL https://linkinghub.elsevier.com/retrieve/pii/S2405844024109590.

[88] Alfio Quarteroni, Gianluigi Rozza, et al. *Reduced order methods for modeling and computational reduction*, volume 9. Springer, 2014.

[89] Gianluigi Rozza, M Hess, G Stabile, M Tezzele, F Ballarin, C Gräßle, M Hinze, S Volkwein, F Chinesta, P Ladeveze, et al. *Volume 2 Snapshot-Based Methods and Algorithms*. De Gruyter, Berlin, Boston, 2021. ISBN 9783110671490. doi: doi:10.1515/9783110671490. URL https://doi.org/10.1515/9783110671490.

[90] Andrea Manzoni, Alfio Quarteroni, and Gianluigi Rozza. Model reduction techniques for fast blood flow simulation in parametrized geometries. *International Journal for Numerical Methods in Biomedical Engineering*, 28(6-7):604–625, June 2012. ISSN 2040-7939, 2040-7947. doi: 10.1002/cnm.1465. URL https://onlinelibrary.wiley.com/doi/10.1002/cnm.1465.

[91] Filippo Salmoiraghi, Francesco Ballarin, Giovanni Corsi, Andrea Mola, Marco Tezzele, Gianluigi Rozza, et al. 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 ECCOMAS Congress 2016, 7th European Conference on Computational Methods in Applied Sciences and Engineering, Crete Island, Greece, June 5-10, 2016*, volume 1, pages 1013–1031. Institute of Structural Analysis and Antiseismic Research School of Civil . . . , 2016.

[92] Anna Ivagnes, Nicola Demo, and Gianluigi Rozza. A shape optimization pipeline for marine propellers by means of reduced order modeling techniques. *International Journal for Numerical Methods in Engineering*, 125(7):e7426, April 2024. ISSN 0029-5981, 1097-0207. doi: 10.1002/nme.7426. URL https://onlinelibrary.wiley.com/doi/10.1002/nme.7426.

[93] Peter Benner, Wil Schilders, Stefano Grivet-Talocia, Alfio Quarteroni, Gianluigi Rozza, and Luís Miguel Silveira. *Model order reduction: volume 3 applications*. De Gruyter, 2020.

[94] Gianluigi Rozza, Haris Malik, Nicola Demo, Marco Tezzele, Michele Girfoglio, Giovanni Stabile, and Andrea Mola. Advances in Reduced Order Methods for parametric industrial problems in computational fluid dynamics. November 2018. doi: 10.48550/arXiv.1811.08319. URL http://arxiv.org/abs/1811.08319. arXiv:1811.08319 [math].

[95] Thomas W Sederberg and Scott R Parry. Free-form deformation of solid geometric models. In *Proceedings of the 13th annual conference on Computer graphics and interactive techniques*, pages 151–160, 1986.

[96] Anwar Koshakji, Alfio Quarteroni, and Gianluigi Rozza. Free Form Deformation techniques applied to 3D shape optimization problems. *Communications in Applied and Industrial Mathematics*, 4, December 2013. ISSN 2038-0909.

[97] Pierfrancesco Siena, Paquale Claudio Africa, Michele Girfoglio, and Gianluigi Rozza. On the accuracy and efficiency of reduced order models: towards real-world applications, September 2024. URL http://arxiv.org/abs/2407.03325. arXiv:2407.03325 [math].

[98] Nicola Demo, Marco Tezzele, and Gianluigi Rozza. A non-intrusive approach for the reconstruction of POD modal coefficients through active subspaces. *Comptes Rendus. Mécanique*, 347(11):873–881, November 2019. ISSN 1873-7234. doi: 10.1016/j.crme.2019.11.012. URL https://comptes-rendus.academie-sciences.fr/mecanique/articles/10.1016/j.crme.2019.11.012/.

[99] Marco Tezzele, Nicola Demo, Giovanni Stabile, Andrea Mola, and Gianluigi Rozza. Enhancing CFD predictions in shape design problems by model and parameter space reduction. *Advanced Modeling and Simulation in Engineering Sciences*, 7(1):40, December 2020. ISSN 2213-7467. doi: 10.1186/s40323-020-00177-y. URL https://amses-journal.springeropen.com/articles/10.1186/s40323-020-00177-y.

[100] Marco Tezzele, Nicola Demo, and Gianluigi Rozza. Shape optimization through proper orthogonal decomposition with interpolation and dynamic mode decomposition enhanced by active subspaces. May 2019. doi: 10.48550/arXiv.1905.05483. URL http://arxiv.org/abs/1905.05483. arXiv:1905.05483 [math].

[101] Guglielmo Padula, Francesco Romor, Giovanni Stabile, and Gianluigi Rozza. Generative models for the deformation of industrial shapes with linear geometric constraints: model order and parameter space reductions, August 2023. URL http://arxiv.org/abs/2308.03662. arXiv:2308.03662 [math].

[102] Francesco Romor, Marco Tezzele, and Gianluigi Rozza. A local approach to parameter space reduction for regression and classification tasks. *Journal of Scientific Computing*, 99 (3):83, June 2024. ISSN 0885-7474, 1573-7691. doi: 10.1007/s10915-024-02542-0. URL https://link.springer.com/10.1007/s10915-024-02542-0.

[103] Paul G Constantine. *Active subspaces: Emerging ideas for dimension reduction in parameter studies*. SIAM, 2015.

[104] Paul G. Constantine, Eric Dow, and Qiqi Wang. Active subspace methods in theory and practice: applications to kriging surfaces. *SIAM Journal on Scientific Computing*, 36(4): A1500–A1524, January 2014. ISSN 1064-8275, 1095-7197. doi: 10.1137/130916138. URL http://arxiv.org/abs/1304.2070. arXiv:1304.2070 [math].

[105] Marco Tezzele, Filippo Salmoiraghi, Andrea Mola, and Gianluigi Rozza. Dimension reduction in heterogeneous parametric spaces with application to naval engineering shape

design problems. *Advanced Modeling and Simulation in Engineering Sciences*, 5(1):25, December 2018. ISSN 2213-7467. doi: 10.1186/s40323-018-0118-3. URL https://amses-journal.springeropen.com/articles/10.1186/s40323-018-0118-3.

[106] Andrea Mola, Marco Tezzele, Mahmoud Gadalla, Federica Valdenazzi, Davide Grassi, Roberta Padovan, and Gianluigi Rozza. Efficient reduction in shape parameter space dimension for ship propeller blade design. May 2019. doi: 10.48550/arXiv.1905.09815. URL http://arxiv.org/abs/1905.09815. arXiv:1905.09815 [cs].

[107] Guglielmo Padula, Francesco Romor, Giovanni Stabile, and Gianluigi Rozza. Generative models for the deformation of industrial shapes with linear geometric constraints: Model order and parameter space reductions. *Computer Methods in Applied Mechanics and Engineering*, 423:116823, April 2024. ISSN 00457825. doi: 10.1016/j.cma.2024.116823. URL https://linkinghub.elsevier.com/retrieve/pii/S0045782524000793.

[108] Guglielmo Padula and Gianluigi Rozza. Generative models for parameter space reduction applied to reduced order modelling, June 2025. URL http://arxiv.org/abs/2506.09721. arXiv:2506.09721 [math].

[109] Steven L Brunton and J Nathan Kutz. *Data-driven science and engineering: Machine learning, dynamical systems, and control*. Cambridge University Press, 2022.

[110] G Berkooz, P Holmes, and J L Lumley. The proper orthogonal decomposition in the analysis of turbulent flows. *Annual Review of Fluid Mechanics*, 25(Volume 25, 1993): 539–575, 1993. ISSN 1545-4479. doi: https://doi.org/10.1146/annurev.fl.25.010193.002543. URL https://www.annualreviews.org/content/journals/10.1146/annurev.fl.25.010193.002543.

[111] Philip J Holmes, John L Lumley, Gal Berkooz, Jonathan C Mattingly, and Ralf W Wittenberg. Low-dimensional models of coherent structures in turbulence. *Physics Reports*, 287(4):337–384, 1997.

[112] Bernd R Noack and Helmut Eckelmann. A low-dimensional Galerkin method for the three-dimensional flow around a circular cylinder. *Physics of Fluids*, 6(1):124–143, 1994.

[113] Saddam Hijazi, Giovanni Stabile, Andrea Mola, and Gianluigi Rozza. Data-driven POD-Galerkin reduced order model for turbulent flows. *Journal of Computational Physics*, 416: 109513, 2020.

[114] Gaetan Kerschen, Jean-claude Golinval, Alexander F Vakakis, and Lawrence A Bergman. The method of proper orthogonal decomposition for dynamical characterization and order reduction of mechanical systems: an overview. *Nonlinear dynamics*, 41:147–169, 2005.

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

[116] Jorge Yáñez Escanciano and Andreas G Class. POD-Galerkin modeling of a heated pool. *Progress in Nuclear Energy*, 113:196–205, 2019.

[117] Francesco Ballarin, Andrea Manzoni, Alfio Quarteroni, and Gianluigi Rozza. Supremizer stabilization of POD-Galerkin approximation of parametrized steady incompressible Navier-Stokes equations. *International Journal for Numerical Methods in Engineering*, 102(5): 1136–1161, 2015.

[118] Giovanni Stabile, Saddam Hijazi, Andrea Mola, Stefano Lorenzi, and Gianluigi Rozza. POD-Galerkin reduced order methods for CFD using finite volume discretisation: vortex shedding around a circular cylinder. *Communications in Applied and Industrial Mathematics*, 8(1):210–236, 2017.

[119] Ivan Prusak, Monica Nonino, Davide Torlo, Francesco Ballarin, and Gianluigi Rozza. An optimisation–based domain–decomposition reduced order model for the incompressible Navier-Stokes equations. *Computers & Mathematics with Applications*, 151:172–189, 2023.

[120] Pierfrancesco Siena, Michele Girfoglio, Francesco Ballarin, and Gianluigi Rozza. Data-driven reduced order modelling for patient-specific hemodynamics of coronary artery bypass grafts with physical and geometrical parameters. *Journal of Scientific Computing*, 94(2):38, 2023.

[121] Dinh Bao Phuong Huynh, David J Knezevic, and Anthony T Patera. A static condensation reduced basis element method: approximation and a posteriori error estimation. *ESAIM: Mathematical Modelling and Numerical Analysis*, 47(1):213–251, 2013.

[122] Chi Hoang, Youngsoo Choi, and Kevin Carlberg. Domain-decomposition least-squares Petrov–Galerkin (DD-LSPG) nonlinear model reduction. *Computer methods in applied mechanics and engineering*, 384:113997, 2021.

[123] Gianluigi Rozza and Karen Veroy. On the stability of the reduced basis method for Stokes equations in parametrized domains. *Computer methods in applied mechanics and engineering*, 196(7):1244–1260, 2007.

[124] Annalisa Buffa, Yvon Maday, Anthony T Patera, Christophe Prud'Homme, and Gabriel Turinici. A priori convergence of the greedy algorithm for the parametrized reduced basis method. *ESAIM: Mathematical modelling and numerical analysis*, 46(3):595–603, 2012.

[125] Sven Kaulmann, Mario Ohlberger, and Bernard Haasdonk. A new local reduced basis discontinuous Galerkin approach for heterogeneous multiscale problems. *Comptes Rendus Mathematique*, 349(23-24):1233–1238, 2011.

[126] Bernard Haasdonk. Convergence rates of the POD–greedy method. *ESAIM: Mathematical modelling and numerical Analysis*, 47(3):859–873, 2013.

[127] Mario Ohlberger and Felix Schindler. Error control for the localized reduced basis multiscale method with adaptive on-line enrichment. *SIAM Journal on Scientific Computing*, 37(6):A2865–A2895, 2015.

[128] Karen Veroy, Christophe Prud'Homme, Dimitrios Rovas, and Anthony Patera. A posteriori error bounds for reduced-basis approximation of parametrized noncoercive and nonlinear elliptic partial differential equations. In *16th AIAA computational fluid dynamics conference*, page 3847, 2003.

[129] Ettore Lappano, Frank Naets, Wim Desmet, Domenico Mundo, Eugene Nijman, et al. A greedy sampling approach for the projection basis construction in parametric model order reduction for structural dynamics models. In *Proceedings of ISMA*, pages 19–21, 2016.

[130] Jiahua Jiang and Yanlai Chen. Adaptive greedy algorithms based on parameter-domain decomposition and reconstruction for the reduced basis method. *International Journal for Numerical Methods in Engineering*, 121(23):5426–5445, 2020.

[131] Wang Chen, Jan S Hesthaven, Bai Junqiang, Yasong Qiu, Zhang Yang, and Yang Tihao. Greedy nonintrusive reduced order model for fluid dynamics. *AIAA Journal*, 56(12):4927–4943, 2018.

[132] Efthymios N Karatzas, Giovanni Stabile, Leo Nouveau, Guglielmo Scovazzi, and Gianluigi Rozza. A reduced basis approach for PDEs on parametrized geometries based on the shifted boundary finite element method and application to a Stokes flow. *Computer Methods in Applied Mechanics and Engineering*, 347:568–587, 2019.

[133] Andrea Ferrero, Angelo Iollo, and Francesco Larocca. Global and local POD models for the prediction of compressible flows with DG methods. *International Journal for Numerical Methods in Engineering*, 116(5):332–357, 2018.

[134] Guosheng Fu and Zhu Wang. POD-(H) DG method for incompressible flow simulations. *Journal of Scientific Computing*, 85(2):24, 2020.

[135] Saray Busto, Giovanni Stabile, Gianluigi Rozza, and M Elena Vázquez-Cendón. POD-Galerkin reduced order methods for combined Navier–Stokes transport equations based on a hybrid FV-FE solver. *Computers & Mathematics with Applications*, 79(2):256–273, 2020.

[136] Sabrina Star, Giovanni Stabile, Gianluigi Rozza, and Joris Degroote. A POD-Galerkin reduced order model of a turbulent convective buoyant flow of sodium over a backward-facing step. *Applied Mathematical Modelling*, 89:486–503, 2021.

[137] Pierfrancesco Siena, Michele Girfoglio, Annalisa Quaini, and Gianluigi Rozza. Stabilized POD reduced order models for convection-dominated incompressible flows. *arXiv preprint arXiv:2404.19600*, 2024.

[138] Francesco Ballarin. Reduced-order models for patient-specific haemodynamics of coronary artery bypass grafts. 2015.

[139] Maxime Barrault, Yvon Maday, Ngoc Cuong Nguyen, and Anthony T Patera. An 'empirical interpolation' method: application to efficient reduced-basis discretization of partial differential equations. *Comptes Rendus Mathematique*, 339(9):667–672, 2004.

[140] Saifon Chaturantabut and Danny C Sorensen. Nonlinear model reduction via discrete empirical interpolation. *SIAM Journal on Scientific Computing*, 32(5):2737–2764, 2010.

[141] Kevin Carlberg, Ray Tuminaro, and Paul Boggs. Preserving Lagrangian structure in nonlinear model reduction with application to structural dynamics. *SIAM Journal on Scientific Computing*, 37(2):B153–B184, 2015.

[142] Stefano Lorenzi, Antonio Cammi, Lelio Luzzi, and Gianluigi Rozza. POD-Galerkin method for finite volume approximation of Navier–Stokes and RANS equations. *Computer Methods in Applied Mechanics and Engineering*, 311:151–179, 2016.

[143] Armin Sheidani, Sajad Salavatidezfouli, Giovanni Stabile, and Gianluigi Rozza. Assessment of URANS and LES methods in predicting wake shed behind a vertical axis wind turbine. *Journal of Wind Engineering and Industrial Aerodynamics*, 232:105285, 2023.

[144] Anna Ivagnes, Giovanni Stabile, Andrea Mola, Gianluigi Rozza, and Traian Iliescu. Data enhanced reduced order methods for turbulent flows. In *Reduction, Approximation, Machine Learning, Surrogates, Emulators and Simulators: RAMSES*, pages 171–198. Springer, 2024.

[145] Peter Benner, Stefano Grivet-Talocia, Alfio Quarteroni, Gianluigi Rozza, Wil Schilders, and Luís Miguel Silveira. *Volume 1 System- and Data-Driven Methods and Algorithms*. De Gruyter, Berlin, Boston, 2021. ISBN 9783110498967. doi: doi:10.1515/9783110498967. URL `https://doi.org/10.1515/9783110498967`.

[146] Haixin Wang, Yadi Cao, Zijie Huang, Yuxuan Liu, Peiyan Hu, Xiao Luo, Zezheng Song, Wanjia Zhao, Jilin Liu, Jinan Sun, et al. Recent advances on machine learning for computational fluid dynamics: A survey. *arXiv preprint arXiv:2408.12171*, 2024.

[147] Bui Thanh Tan. Proper orthogonal decomposition extensions and their applications in steady aerodynamics. *Master of Engineering in High Performance Computation for Engineered Systems (HPCES), Singapore-MIT Alliance*, 2003.

[148] Nicola Demo, Marco Tezzele, and Gianluigi Rozza. EZyRB: Easy reduced basis method. *Journal of Open Source Software*, 3(24):661, 2018.

[149] Fabrizio Garotta, Nicola Demo, Marco Tezzele, Massimo Carraturo, Alessandro Reali, and Gianluigi Rozza. Reduced order isogeometric analysis approach for PDEs in parametrized domains. In *Quantification of Uncertainty: Improving Efficiency and Technology: QUIET Selected Contributions*, pages 153–170. Springer, 2020.

[150] Filippo Salmoiraghi, Angela Scardigli, Haysam Telib, and Gianluigi Rozza. Free-form deformation, mesh morphing and reduced-order methods: enablers for efficient aerodynamic shape optimisation. *International Journal of Computational Fluid Dynamics*, 32(4-5):233–247, 2018.

[151] Marco Tezzele, Nicola Demo, Andrea Mola, and Gianluigi Rozza. An integrated data-driven computational pipeline with model order reduction for industrial and applied mathematics. In *Novel mathematics inspired by industrial challenges*, pages 179–200. Springer, 2022.

[152] D. Xiao, F. Fang, C.E. Heaney, I.M. Navon, and C.C. Pain. A domain decomposition method for the non-intrusive reduced order modelling of fluid flow. *Computer Methods in Applied Mechanics and Engineering*, 354:307–330, 2019. ISSN 0045-7825.

[153] Dunhui Xiao, Fangxin Fang, Claire E Heaney, IM Navon, and CC Pain. A domain decomposition method for the non-intrusive reduced order modelling of fluid flow. *Computer Methods in Applied Mechanics and Engineering*, 354:307–330, 2019.

[154] Marco Fossati and Wagdi G Habashi. Multiparameter analysis of aero-icing problems using proper orthogonal decomposition and multidimensional interpolation. *AIAA journal*, 51(4):946–960, 2013.

[155] Mengwu Guo and Jan S Hesthaven. Reduced order modeling for nonlinear structural analysis using Gaussian process regression. *Computer methods in applied mechanics and engineering*, 341:807–826, 2018.

[156] Giulio Ortali, Nicola Demo, Gianluigi Rozza, et al. A Gaussian process regression approach within a data-driven POD framework for engineering problems in fluid dynamics. *Mathematics in Engineering*, 4(3):1–16, 2022.

[157] Qian Wang, Jan S Hesthaven, and Deep Ray. Non-intrusive reduced order modeling of unsteady flows using artificial neural networks with application to a combustion problem. *Journal of computational physics*, 384:289–307, 2019.

[158] Dunhui Xiao, Fangxin Fang, Christopher Pain, and Guangwei Hu. Non-intrusive reduced-order modelling of the Navier–Stokes equations based on RBF interpolation. *International Journal for Numerical Methods in Fluids*, 79(11):580–595, 2015.

[159] Arash Hajisharifi, Michele Girfoglio, Annalisa Quaini, and Gianluigi Rozza. A comparison of data-driven reduced order models for the simulation of mesoscale atmospheric flow. *Finite Elements in Analysis and Design*, 228:104050, 2024.

[160] Matteo Ripepi, Mark Johannes Verveld, NW Karcher, Thomas Franz, Mohammad Abu-Zurayk, Stefan Görtz, and Thiemo M Kier. Reduced-order models for aerodynamic applications, loads and MDO. *CEAS Aeronautical Journal*, 9(1):171–193, 2018.

[161] Dunhui Xiao, Pan Yang, Fangxin Fang, Jiansheng Xiang, Chris C Pain, and Ionel M Navon. Non-intrusive reduced order modelling of fluid–structure interactions. *Computer Methods in Applied Mechanics and Engineering*, 303:35–54, 2016.

[162] Alberto Sartori, Antonio Cammi, Lelio Luzzi, and Gianluigi Rozza. A multi-physics reduced order model for the analysis of lead fast reactor single channel. *Annals of Nuclear Energy*, 87:198–208, 2016.

[163] Majdi Ibrahim Ahmad Radaideh. *A novel framework for data-driven modeling, uncertainty quantification, and deep learning of nuclear reactor simulations*. PhD thesis, University of Illinois at Urbana-Champaign, 2019.

[164] Fahad Alsayyari, Marco Tiberga, Zoltán Perkó, Danny Lathouwers, and Jan Leen Kloosterman. A non-intrusive adaptive reduced order modeling approach for a molten salt reactor system. *Annals of Nuclear Energy*, 141:107321, 2020.

[165] Qian Wang, Jan S. Hesthaven, and Deep Ray. Non-intrusive reduced order modeling of unsteady flows using artificial neural networks with application to a combustion problem. *Journal of Computational Physics*, 384:289–307, May 2019. ISSN 00219991. doi: 10.1016/j.jcp.2019.01.031.

[166] George Em Karniadakis, Ioannis G Kevrekidis, Lu Lu, Paris Perdikaris, Sifan Wang, and Liu Yang. Physics-informed machine learning. *Nature Reviews Physics*, 3(6):422–440, 2021.

[167] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016.

[168] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436–444, 2015.

[169] Yann LeCun. *Modeles connexionnistes de l'apprentissage (connectionist learning models)*. PhD thesis, Universit
'e P. et M. Curie (Paris 6), June 1987.

[170] Claire E Heaney, Zef Wolffs, Jón Atli Tómasson, Lyes Kahouadji, Pablo Salinas, André Nicolle, Ionel M Navon, Omar K Matar, Narakorn Srinil, and Christopher C Pain. An AI-based non-intrusive reduced-order model for extended domains applied to multiphase flow in pipes. *Physics of Fluids*, 34(5), 2022.

[171] Jianqing Huang, Hecong Liu, and Weiwei Cai. Online in situ prediction of 3-D flame evolution from its history 2-D projections via deep learning. *Journal of Fluid Mechanics*, 875:R2, 2019.

[172] Maziar Raissi, Paris Perdikaris, and George E Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving

nonlinear partial differential equations. *Journal of Computational physics*, 378:686–707, 2019.

[173] Arthur Feeney, Zitong Li, Ramin Bostanabad, and Aparna Chandramowlishwaran. 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*, pages 1–15, 2023.

[174] Nikola Kovachki, Zongyi Li, Burigede Liu, Kamyar Azizzadenesheli, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Neural operator: Learning maps between function spaces with applications to PDEs. *Journal of Machine Learning Research*, 24(89):1–97, 2023.

[175] Yuanhong Chen, Yifan Lin, Xiang Sun, Chunxin Yuan, and Zhen Gao. Tensor decomposition-based neural operator with dynamic mode decomposition for parameterized time-dependent problems. *Journal of Computational Physics*, 533:113996, 2025.

[176] Steven L Brunton, Joshua L Proctor, and J Nathan Kutz. Sparse identification of nonlinear dynamics with control (SINDYc). *IFAC-PapersOnLine*, 49(18):710–715, 2016.

[177] Steven L Brunton, Joshua L Proctor, and J Nathan Kutz. Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *Proceedings of the national academy of sciences*, 113(15):3932–3937, 2016.

[178] Han Bao, Nam Dinh, and Linyu Lin. Machine learning in nuclear thermal hydraulics, part 2: Opportunities and perspectives. Technical report, Idaho National Laboratory (INL), Idaho Falls, ID (United States), 2020.

[179] Jordan Andrew Melendez, Christian Drischler, RJ Furnstahl, AJ Garcia, and Xilin Zhang. Model reduction methods for nuclear emulators. *Journal of Physics G: Nuclear and Particle Physics*, 49(10):102001, 2022.

[180] Arsen S Iskhakov, Cheng-Kai Tai, Igor A Bolotnov, and Nam T Dinh. A perspective on data-driven coarse grid modeling for system-level thermal hydraulics. *Nuclear Science and Engineering*, 197(10):2527–2542, 2023.

[181] Qiao Hong, Ma Jun, Wang Bo, Tan Sichao, Zhang Jiayi, Liang Biao, Li Tong, and Tian Ruifeng. Application of data-driven technology in nuclear engineering: Prediction, classification and design optimization. *Annals of Nuclear Energy*, 194:110089, 2023.

[182] Dennis P Prill, Markus Stokmaier, and Andreas G Class. Methodology of roming applying POD of BWR fuel assemblies applied to introductory examples. *PAMM*, 12(1):237–238, 2012.

[183] Dennis P Prill and Andreas G Class. Non-linear proper orthogonal decomposition reduced order modeling: Bifurcation study of a natural convection circuit. *PAMM*, 13(1):285–286, 2013.

[184] Dennis Paul Prill. *A non-linear reduced order methodology applicable to boiling water reactor stability analysis*. PhD Thesis, Karlsruhe, Karlsruher Institut für Technologie (KIT), Diss., 2013, 2013. URL `https://d-nb.info/104636264X/34`.

[185] Dennis P Prill and Andreas G Class. Verification of nonlinear proper orthogonal decomposition reduced order modeling for BWR fuel assemblies. In *International Conference on Nuclear Engineering*, volume 55812, page V004T09A088. American Society of Mechanical Engineers, 2013.

[186] DP Prill and AG Class. Semi-automated proper orthogonal decomposition reduced order model non-linear analysis for future BWR stability. *Annals of Nuclear Energy*, 67:70–90, 2014.

[187] M Zarei. On a reduced order modeling of the nuclear reactor dynamics. *Applied Mathematics and Computation*, 393:125819, 2021.

[188] Qi Lu, Yuan Yuan, Fengchen Li, Bo Yang, Zhe Li, Yu Ma, Yiyu Gu, and Dingming Liu. Prediction method for thermal-hydraulic parameters of nuclear reactor system based on deep learning algorithm. *Applied Thermal Engineering*, 196:117272, 2021.

[189] Young Suk Bang, Jaeseok Heo, Ha Neul Na, and Kyoung-Ho Kang. Reduced order models for thermal hydraulic transient analysis in nuclear engineering. *Annals of Nuclear Energy*, 211:111002, 2025.

[190] Péter German, Mauricio Tano, Carlo Fiorina, and Jean C Ragusa. GeN-ROM—an OpenFOAM®-based multiphysics reduced-order modeling framework for the analysis of molten salt reactors. *Progress in Nuclear Energy*, 146:104148, 2022.

[191] Carlo Fiorina, Ivor Clifford, Manuele Aufiero, and Konstantin Mikityuk. GeN-Foam: a novel OpenFOAM® based multi-physics solver for 2D/3D transient analysis of nuclear reactors. *Nuclear Engineering and Design*, 294:24–37, 2015.

[192] Peter German, Jean C Ragusa, and Carlo Fiorina. Application of multiphysics model order reduction to doppler/neutronic feedback. *EPJ Nuclear Sciences & Technologies*, 5:17, 2019.

[193] Peter German. *Multiphysics Model Order Reduction for Molten Salt Reactors*. PhD thesis, 2021.

[194] Rabab Elzohery and Jeremy Roberts. Exploring transient, neutronic, reduced-order models using DMD/POD-Galerkin and data-driven DMD. *EPJ Web of Conferences*, 247:15019, 2021. ISSN 2100-014X. doi: 10.1051/epjconf/202124715019.

[195] Rabab Elzohery and Jeremy Roberts. Modeling neutronic transients with Galerkin projection onto a greedy-sampled, POD subspace. *Annals of Nuclear Energy*, 162:108487, November 2021. ISSN 0306-4549. doi: 10.1016/j.anucene.2021.108487.

[196] Rabab Elzohery and Jeremy Roberts. Application of model-order reduction of nonlinear time-dependent neutronics via POD-Galerkin projection and matrix discrete empirical interpolation. *Annals of Nuclear Energy*, 179:109396, December 2022. ISSN 0306-4549. doi: 10.1016/j.anucene.2022.109396.

[197] Dongli Huang, Hany Abdel-Khalik, Cristian Rabiti, and Frederick Gleicher. Dimensionality reducibility for multi-physics reduced order modeling. *Annals of Nuclear Energy*, 110: 526–540, 2017.

[198] Helin Gong, Sibo Cheng, Zhang Chen, and Qing Li. Data-enabled physics-informed machine learning for reduced-order modeling digital twin: application to nuclear reactor physics. *Nuclear Science and Engineering*, 196(6):668–693, 2022.

[199] Huilun Kang, Zhaofei Tian, Guangliang Chen, Lei Li, and Tianyu Wang. Application of POD reduced-order algorithm on data-driven modeling of rod bundle. *Nuclear Engineering and Technology*, 54(1):36–48, 2022.

[200] Yijun Zhang, Wenhuai Li, Sitao Peng, Jinggang Li, Ting Wang, Qingyun He, Tao Wang, Haoliang Lu, and Ling Zeng. Enhancing uncertainty analysis: POD-DNNs for reduced order modeling of neutronic transient behavior. 435:113969. ISSN 0029-5493. doi: 10.1016/j.nucengdes.2025.113969. URL https://www.sciencedirect.com/science/article/pii/S0029549325001463.

[201] Wenhuai Li, Jinggang Li, Jianfan Yao, Sitao Peng, Qingyun He, Tao Wang, Yijun Zhang, Haoliang Lu, and Ting Wang. Mode decomposition of core dynamics transients using higher-order DMD method. 427:113417, . ISSN 00295493. doi: 10.1016/j.nucengdes.2024.113417. URL https://linkinghub.elsevier.com/retrieve/pii/S002954932400517X.

[202] Wenhuai Li, Sitao Peng, Jinggang Li, Qingyun He, Tao Wang, Yijun Zhang, Haoliang Lu, Ting Wang, and Peng Ding. Prediction of state transitions in 3D core dynamics and xenon transients based on dynamic mode decomposition. 197:110258, . ISSN 03064549. doi: 10.1016/j.anucene.2023.110258. URL https://linkinghub.elsevier.com/retrieve/pii/S0306454923005777.

[203] Kurt A. Dominesey and Wei Ji. Reduced-order modeling of neutron transport separated in axial and radial space by Proper Generalized Decomposition with applications to nuclear reactor physics. 213:111162. ISSN 0306-4549. doi: 10.1016/j.anucene.2024.111162. URL https://www.sciencedirect.com/science/article/pii/S0306454924008259.

[204] Hanxing Liu and Han Zhang. A reduced order model based on ANN-POD algorithm for steady-state neutronics and thermal-hydraulics coupling problem. 2023(1):9385756. ISSN 1687-6083. doi: 10.1155/2023/9385756. URL https://onlinelibrary.wiley.com/doi/abs/10.1155/2023/9385756.

[205] Junda Zhang, Yinuo Chen, Xiaojing Liu, Xiang Chai, Hui He, Jinbiao Xiong, and Tengfei Zhang. Optimizing thermal stress distribution in heat-pipe-cooled micro-reactors using multi-physics data-driven methods. *Energy*, 323:135679, 2025.

[206] Stefano Riva, Carolina Introini, and Antonio Cammi. Multi-physics model bias correction with data-driven reduced order techniques: Application to nuclear case studies. *Applied Mathematical Modelling*, 135:243–268, November 2024. ISSN 0307-904X. doi: 10.1016/j.apm.2024.06.040. URL https://linkinghub.elsevier.com/retrieve/pii/S0307904X24003196. Publisher: Elsevier BV.

[207] Stefano Riva. *Reduced basis methods for data assimilation in real thermal hydraulics systems*. PhD Thesis, Politecnico di Milano, Milan, 2020. URL https://www.politesi.polimi.it/handle/10589/181556.

[208] Stefano Riva, Carolina Introini, Stefano Lorenzi, and Antonio Cammi. Hybrid Data Assimilation methods, Part II: Application to the DYNASTY experimental facility. *Annals of Nuclear Energy*, 190:109863, September 2023. ISSN 0306-4549. doi: 10.1016/j.anucene.2023.109863. URL https://linkinghub.elsevier.com/retrieve/pii/S0306454923001822. Publisher: Elsevier BV.

[209] Stefano Riva, Carolina Introini, Stefano Lorenzi, and Antonio Cammi. Hybrid data assimilation methods, Part I: Numerical comparison between GEIM and PBDW. *Annals of Nuclear Energy*, 190:109864, September 2023. ISSN 0306-4549. doi: 10.1016/j.anucene.2023.109864. URL https://linkinghub.elsevier.com/retrieve/pii/S0306454923001834. Publisher: Elsevier BV.

[210] Carolina Introini, Stefano Riva, Stefano Lorenzi, Simone Cavalleri, and Antonio Cammi. Non-intrusive system state reconstruction from indirect measurements: A novel approach based on Hybrid Data Assimilation methods. *Annals of Nuclear Energy*, 182:109538, March 2023. ISSN 0306-4549. doi: 10.1016/j.anucene.2022.109538. URL https://linkinghub.elsevier.com/retrieve/pii/S0306454922005680. Publisher: Elsevier BV.

[211] Antonio Cammi, Stefano Riva, Carolina Introini, Lorenzo Loi, and Enrico Padovani. Data-driven model order reduction for sensor positioning and indirect reconstruction with noisy data: Application to a Circulating Fuel Reactor. *Nuclear Engineering and Design*, 421:113105, May 2024. ISSN 0029-5493. doi: 10.1016/j.nucengdes.2024.113105. URL https://linkinghub.elsevier.com/retrieve/pii/S002954932400205X. Publisher: Elsevier BV.

[212] Mauricio E Tano and Pablo Rubiolo. Development of explainable data-driven turbulence models with application to liquid fuel nuclear reactors. *Energies*, 15(19):6861, 2022.

[213] Huilun Kang, Zhaofei Tian, Guangliang Chen, Lei Li, and Tianhui Chu. Investigation on the non-intrusive multi-fidelity reduced-order modeling for PWR rod bundles. *Nuclear Engineering and Technology*, 54(5):1825–1834, 2022.

[214] Andreas Buhr, Laura Iapichino, Mario Ohlberger, Stephan Rave, Felix Schindler, and Kathrin Smetana. Localized model reduction for parameterized problems. In *Handbook on Model Order Reduction*. Walter De Gruyter, 2020.

[215] Alexander Heinlein, Axel Klawonn, Martin Lanser, and Janine Weber. Combining machine learning and domain decomposition methods for the solution of partial differential equations—a review. *GAMM-Mitteilungen*, 44(1):e202100001, 2021.

[216] Alf Emil Løvgren, Yvon Maday, and Einar M Rønquist. The reduced basis element method for fluid flows. In *Analysis and simulation of fluid dynamics*, pages 129–154. Springer, 2006.

[217] Paola F Antonietti, Paolo Pacciarini, and Alfio Quarteroni. A discontinuous Galerkin reduced basis element method for elliptic problems. *ESAIM: Mathematical Modelling and Numerical Analysis*, 50(2):337–360, 2016.

[218] Sébastien Riffaud, Michel Bergmann, Charbel Farhat, Sebastian Grimberg, and Angelo Iollo. The DGDD method for reduced-order modeling of conservation laws. *Journal of Computational Physics*, 437:110336, 2021.

[219] Giulia Sambataro. *Component-based model order reduction procedures for large scale THM systems*. PhD thesis, Bordeaux, 2022.

[220] Simone Deparis, Davide Forti, Paola Gervasio, and Alfio Quarteroni. INTERNODES: an accurate interpolation-based method for coupling the Galerkin solutions of PDEs on subdomains featuring non-conforming interfaces. *Computers & Fluids*, 141: 22–41, December 2016. ISSN 00457930. doi: 10.1016/j.compfluid.2016.03.033. URL `https://linkinghub.elsevier.com/retrieve/pii/S0045793016300937`.

[221] Luca Pegolotti, Martin R Pfaller, Alison L Marsden, and Simone Deparis. Model order reduction of flow based on a modular geometrical approximation of blood vessels. *Computer methods in applied mechanics and engineering*, 380:113762, 2021.

[222] Sébastien Riffaud. *Reduced-order models: convergence between scientific computing and data for fluid mechanics*. PhD thesis, Université de Bordeaux, 2020.

[223] Francesco Ballarin and Gianluigi Rozza. POD-Galerkin monolithic reduced order models for parametrized fluid-structure interaction problems. *International Journal for Numerical Methods in Fluids*, 82(12):1010–1034, December 2016. ISSN 0271-2091, 1097-0363.

doi: 10.1002/fld.4252. URL https://onlinelibrary.wiley.com/doi/10.1002/fld.4252. Publisher: Wiley.

[224] Monica Nonino. *On the application of the Reduced Basis Method to Fluid–Structure Interaction problems*. PhD Thesis, SISSA, Trieste, Italy, 2020.

[225] Angelo Iollo, Giulia Sambataro, and Tommaso Taddei. A one-shot overlapping schwarz method for component-based model reduction: application to nonlinear elasticity. *Computer Methods in Applied Mechanics and Engineering*, 404:115786, 2023.

[226] Patrick Henning, Axel Målqvist, and Daniel Peterseim. A localized orthogonal decomposition method for semi-linear elliptic problems. *ESAIM: Mathematical Modelling and Numerical Analysis*, 48(5):1331–1349, 2014.

[227] Niccolò Discacciati and Jan S Hesthaven. Localized model order reduction and domain decomposition methods for coupled heterogeneous systems. *International Journal for Numerical Methods in Engineering*, 124(18):3964–3996, 2023.

[228] Niccolò Discacciati and Jan S Hesthaven. Model reduction of coupled systems based on non-intrusive approximations of the boundary response maps. *Computer Methods in Applied Mechanics and Engineering*, 420:116770, 2024.

[229] Kathrin Smetana and Tommaso Taddei. Localized model reduction for nonlinear elliptic partial differential equations: localized training, partition of unity, and adaptive enrichment. *SIAM Journal on Scientific Computing*, 45(3):A1300–A1331, 2023.

[230] Tommaso Taddei, Xuejun Xu, and Lei Zhang. A non-overlapping optimization-based domain decomposition approach to component-based model reduction of incompressible flows. *Journal of Computational Physics*, 509:113038, 2024.

[231] Yvon Maday and Einar M Rønquist. A reduced-basis element method. *Journal of scientific computing*, 17:447–459, 2002.

[232] Yvon Maday and Einar M Ronquist. The reduced basis element method: application to a thermal fin problem. *SIAM Journal on Scientific Computing*, 26(1):240–258, 2004.

[233] Alf Emil Løvgren, Yvon Maday, and Einar M Rønquist. A reduced basis element method for the steady Stokes problem. *ESAIM: Mathematical Modelling and Numerical Analysis*, 40(3):529–552, 2006.

[234] Yanlai Chen, Jan S Hesthaven, and Yvon Maday. 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*, pages 141–152. Springer, 2011.

[235] Laura Iapichino, Alfio Quarteroni, and Gianluigi Rozza. A reduced basis hybrid method for the coupling of parametrized domains represented by fluidic networks. *Computer Methods in Applied Mechanics and Engineering*, 221:63–82, 2012.

[236] Laura Iapichino. *Reduced basis methods for the solution of parametrized PDEs in repetitive and complex networks with application to CFD*. PhD thesis, EPFL, 2012.

[237] Immanuel Martini, Gianluigi Rozza, and Bernard Haasdonk. Reduced basis approximation and a-posteriori error estimation for the coupled Stokes-Darcy system. *Advances in Computational Mathematics*, 41:1131–1157, 2015.

[238] JL Eftang, DBP Huynh, DJ Knezevic, EM Ronquist, and AT Patera. Adaptive port reduction in static condensation. *IFAC Proceedings Volumes*, 45(2):695–699, 2012.

[239] Dinh Bao Phuong Huynh, David J Knezevic, and Anthony T Patera. A static condensation reduced basis element method: Complex problems. *Computer Methods in Applied Mechanics and Engineering*, 259:197–216, 2013.

[240] Pierre Kerfriden, Olivier Goury, Timon Rabczuk, and Stephane Pierre-Alain Bordas. A partitioned model order reduction approach to rationalise computational expenses in nonlinear fracture mechanics. *Computer methods in applied mechanics and engineering*, 256:169–188, 2013.

[241] Sylvain Vallaghé and Anthony T Patera. The static condensation reduced basis element method for a mixed-mean conjugate heat exchanger model. *SIAM Journal on Scientific Computing*, 36(3):B294–B320, 2014.

[242] Lin Mu and Guannan Zhang. A domain decomposition model reduction method for linear convection-diffusion equations with random coefficients. *SIAM Journal on Scientific Computing*, 41(3):A1984–A2011, 2019.

[243] Amina Benaceur and Anthony Patera. Port-reduced reduced-basis component method for steady state Navier–Stokes and passive scalar equations. 2022.

[244] Paolo Pacciarini, Paola Gervasio, and Alfio Quarteroni. Spectral based discontinuous Galerkin reduced basis element method for parametrized Stokes problems. *Computers & Mathematics with Applications*, 72(8):1977–1987, 2016.

[245] Seung Whan Chung, Youngsoo Choi, Pratanu Roy, Thomas Moore, Thomas Roy, Tiras Y Lin, Du T Nguyen, Christopher Hahn, Eric B Duoss, and Sarah E Baker. Train small, model big: Scalable physics simulators via reduced order modeling and domain decomposition. *Computer Methods in Applied Mechanics and Engineering*, 427:117041, 2024.

[246] Max Jensen. *Discontinuous Galerkin methods for Friedrichs systems with irregular solutions*. PhD thesis, University of Oxford, 2004.

[247] Francesco Romor, Davide Torlo, and Gianluigi Rozza. Friedrichs' systems discretized with the DGM: domain decomposable model order reduction and Graph Neural Networks approximating vanishing viscosity solutions. *Journal of Computational Physics*, page 113915, 2025.

[248] Nirav Vasant Shah, Martin Wilfried Hess, and Gianluigi Rozza. Discontinuous Galerkin model order reduction of geometrically parametrized Stokes equation. In *Numerical Mathematics and Advanced Applications ENUMATH 2019: European Conference, Egmond aan Zee, The Netherlands, September 30-October 4*, pages 551–561. Springer, 2020.

[249] Kun Li, Ting-Zhu Huang, Liang Li, and Stéphane Lanteri. Simulation of the interaction of light with 3-D metallic nanostructures using a proper orthogonal decomposition-Galerkin reduced-order discontinuous Galerkin time-domain method. *Numerical Methods for Partial Differential Equations*, 39(2):932–954, 2023.

[250] Tim Keil, Mario Ohlberger, and Felix Schindler. Adaptive localized reduced basis methods for large scale PDE-constrained optimization. In *International Conference on Large-Scale Scientific Computing*, pages 108–116. Springer, 2023.

[251] Felix Schindler, B. Haasdonk, Mario Ohlberger, and Sven Kaulmann. The localized reduced basis multiscale method. volume 9, pages 393–403, 09 2012.

[252] Sven Kaulmann, Bernd Flemisch, Bernard Haasdonk, K-A Lie, and Mario Ohlberger. The localized reduced basis multiscale method for two-phase flows in porous media. *International Journal for Numerical Methods in Engineering*, 102(5):1018–1040, 2015.

[253] Mario Ohlberger, Stephan Rave, and Felix Schindler. True error control for the localized reduced basis method for parabolic problems. *Model reduction of parametrized systems*, pages 169–182, 2017.

[254] Jens M Melenk and Ivo Babuška. The partition of unity finite element method: basic theory and applications. *Computer methods in applied mechanics and engineering*, 139 (1-4):289–314, 1996.

[255] Ivo Babuška and Jens M Melenk. The partition of unity method. *International journal for numerical methods in engineering*, 40(4):727–758, 1997.

[256] Ivo Babuska and Robert Lipton. Optimal local approximation spaces for generalized finite element methods with application to multiscale problems. *Multiscale Modeling & Simulation*, 9(1):373–406, 2011.

[257] Ivo Babuška, Robert Lipton, Paul Sinz, and Michael Stuebner. Multiscale-spectral GFEM and optimal oversampling. *Computer Methods in Applied Mechanics and Engineering*, 364: 112960, 2020.

[258] Julia Schleuß and Kathrin Smetana. Optimal local approximation spaces for parabolic problems. *Multiscale Modeling & Simulation*, 20(1):551–582, 2022.

[259] Joan Baiges, Ramon Codina, and Sergio Idelsohn. A domain decomposition strategy for reduced order models. application to the incompressible Navier–Stokes equations. *Computer Methods in Applied Mechanics and Engineering*, 267:23–42, 2013.

[260] Y. Efendiev, T. Y. Hou, and V. Ginting. Multiscale finite element methods for nonlinear problems and their applications. *Communications in Mathematical Sciences*, 2(4):553–589, 2004.

[261] Yalchin Efendiev and T23224321112 Hou. Multiscale finite element methods for porous media flows and their applications. *Applied Numerical Mathematics*, 57(5-7):577–596, 2007.

[262] Yalchin Efendiev and Thomas Y Hou. *Multiscale finite element methods: theory and applications*, volume 4. Springer Science & Business Media, 2009.

[263] Yalchin Efendiev, Juan Galvis, and Thomas Y Hou. Generalized multiscale finite element methods (GMsFEM). *Journal of computational physics*, 251:116–135, 2013.

[264] Eric T Chung, Yalchin Efendiev, and Wing Tat Leung. Generalized multiscale finite element methods for wave propagation in heterogeneous media. *Multiscale Modeling & Simulation*, 12(4):1691–1721, 2014.

[265] Yalchin Efendiev, Guanglian Li, Michael Presho, et al. Generalized multiscale finite element methods: Oversampling strategies. *International Journal for Multiscale Computational Engineering*, 12(6), 2014.

[266] Patrick Henning and Axel Målqvist. Localized orthogonal decomposition techniques for boundary value problems. *SIAM Journal on Scientific Computing*, 36(4):A1609–A1634, 2014.

[267] Eric Chung, Yalchin Efendiev, and Thomas Y Hou. Adaptive multiscale model reduction with generalized multiscale finite element methods. *Journal of Computational Physics*, 320: 69–95, 2016.

[268] Ivan Prusak. *Application of optimisation-based domain–decomposition reduced order models to parameter-dependent fluid dynamics and multiphysics problems*. PhD thesis, SISSA, 2023.

[269] Ivan Prusak, Davide Torlo, Monica Nonino, and Gianluigi Rozza. Optimisation–based coupling of finite element model and reduced order model for computational fluid dynamics. *arXiv preprint arXiv:2402.10570*, 2024.

[270] Monica Nonino, Francesco Ballarin, and Gianluigi Rozza. A monolithic and a partitioned, reduced basis method for fluid-structure interaction problems. *Fluids*, 6(6):229, June 2021. ISSN 2311-5521. doi: 10.3390/fluids6060229. URL https://www.mdpi.com/2311-5521/6/6/229. Publisher: MDPI AG.

[271] Monica Nonino, Francesco Ballarin, Gianluigi Rozza, and Yvon Maday. Overcoming slowly decaying Kolmogorov n-width by transport maps: application to model order reduction of fluid dynamics and fluid–structure interaction problems. *Advances in Computational Science and Engineering*, 1(1):36–58, 2023. ISSN 2837-1739. doi: 10.3934/acse.2023002. URL http://arxiv.org/abs/1911.06598. arXiv:1911.06598 [math].

[272] Monica Nonino, Francesco Ballarin, Gianluigi Rozza, and Yvon Maday. A reduced basis method by means of transport maps for a fluid-structure interaction problem with slowly decaying Kolmogorov n-width. *Advances in Computational Science & Engineering (ACSE)*, 1(1), 2023. URL https://www.aimsciences.org/data/article/export-pdf?doi=10.3934/acse.2023002.

[273] Moaad Khamlich, Federico Pichi, and Gianluigi Rozza. Model order reduction for bifurcating phenomena in fluid-structure interaction problems. *International Journal for Numerical Methods in Fluids*, 94(10):1611–1640, October 2022. ISSN 0271-2091, 1097-0363. doi: 10.1002/fld.5118. URL https://onlinelibrary.wiley.com/doi/10.1002/fld.5118.

[274] T. Lieu, C. Farhat, and M. Lesoinne. Reduced-order fluid/structure modeling of a complete aircraft configuration. *Computer Methods in Applied Mechanics and Engineering*, 195(41-43):5730–5742, August 2006. ISSN 0045-7825. doi: 10.1016/j.cma.2005.08.026. URL https://linkinghub.elsevier.com/retrieve/pii/S0045782505005153. Publisher: Elsevier BV.

[275] C.M. Colciago, S. Deparis, and A. Quarteroni. Comparisons between reduced order models and full 3D models for fluid-structure interaction problems in haemodynamics. *Journal of Computational and Applied Mathematics*, 265:120–138, August 2014. ISSN 0377-0427. doi: 10.1016/j.cam.2013.09.049. URL https://linkinghub.elsevier.com/retrieve/pii/S0377042713005049. Publisher: Elsevier BV.

[276] Andrea Manzoni, Diana Bonomi, and Alfio Quarteroni. Reduced order modeling for cardiac electrophysiology and mechanics: New methodologies, challenges and perspectives. *Mathematical and numerical modeling of the cardiovascular system and applications*, pages 115–166, 2018.

[277] Ludovica Cicci, Stefania Fresca, Stefano Pagani, Andrea Manzoni, Alfio Quarteroni, et al. Projection-based reduced order models for parameterized nonlinear time-dependent problems arising in cardiac mechanics. *Mathematics in Engineering*, 5(2):1–38, 2023.

[278] Alberto Corigliano, Martino Dossi, Stefano Mariani, et al. 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)*, pages 4115–4123, 2014.

[279] Alberto Corigliano, Martino Dossi, and Stefano Mariani. Model order reduction and domain decomposition strategies for the solution of the dynamic elastic–plastic structural problem. *Computer Methods in Applied Mechanics and Engineering*, 290:127–155, 2015.

[280] Martino Dossi. *Combined Model Order Reduction and Domain Decomposition strategies for the solution of non-linear and multi-physics structural problems*. PhD Thesis, Politecnico di Milano, Milan, January 2015. URL `https://tesidottorato.depositolegale.it/static/PDF/web/viewer.jsp`.

[281] E. Liberge and A. Hamdouni. Reduced order modelling method via proper orthogonal decomposition (POD) for flow around an oscillating cylinder. *Journal of Fluids and Structures*, 26(2):292–311, February 2010. ISSN 0889-9746. doi: 10.1016/j.jfluidstructs.2009.10.006. URL `https://linkinghub.elsevier.com/retrieve/pii/S0889974609001352`. Publisher: Elsevier BV.

[282] Paola Gervasio and Alfio Quarteroni. Analysis of the INTERNODES method for non-conforming discretizations of elliptic equations. *Computer Methods in Applied Mechanics and Engineering*, 334:138–166, 2018.

[283] Elena Zappon, Andrea Manzoni, and Alfio Quarteroni. A staggered-in-time and non-conforming-in-space numerical framework for realistic cardiac electrophysiology outputs, August 2023. URL `http://arxiv.org/abs/2308.03884`. arXiv:2308.03884 [math].

[284] Elena Zappon, Andrea Manzoni, and Alfio Quarteroni. Efficient and certified solution of parametrized one-way coupled problems through DEIM-based data projection across non-conforming interfaces. *Advances in Computational Mathematics*, 49(2):21, April 2023. ISSN 1019-7168, 1572-9044. doi: 10.1007/s10444-022-10008-w. URL `https://link.springer.com/10.1007/s10444-022-10008-w`.

[285] Alfio Quarteroni and Alberto Valli. *Domain decomposition methods for partial differential equations*. Oxford University Press, 1999.

[286] Marcelo Buffoni, Haysam Telib, and Angelo Iollo. Iterative methods for model reduction by domain decomposition. *Computers & Fluids*, 38(6):1160–1167, 2009.

[287] D Cinquegrana, RS Donelli, and A Viviani. A hybrid method based on POD and domain decomposition to compute the 2-D aerodynamics flow field. *AIMETA, Bologna, Italy*, pages 1–10, 2011.

[288] Junpeng Song and Hongxing Rui. A reduced-order Schwarz domain decomposition method based on POD for the convection-diffusion equation. *Computers & Mathematics with Applications*, 160:60–69, 2024.

[289] Marco Discacciati, Ben J Evans, and Matteo Giacomini. An overlapping domain decomposition method for the solution of parametric elliptic problems via proper generalized decomposition. *Computer Methods in Applied Mechanics and Engineering*, 418:116484, 2024.

[290] Christopher R Wentland, Francesco Rizzi, Joshua Barnett, and Irina Tezaur. The role of interface boundary conditions and sampling strategies for Schwarz-based coupling of projection-based reduced order models. *arXiv preprint arXiv:2410.04668*, 2024.

[291] I Maier and B Haasdonk. An iterative domain decomposition procedure for the reduced-basis-method. Master's thesis, University of Stuttgart, 2011.

[292] Immanuel Maier and Bernard Haasdonk. A Dirichlet–Neumann reduced basis method for homogeneous domain decomposition problems. *Applied Numerical Mathematics*, 78: 31–48, 2014.

[293] Joshua Barnett, Irina Tezaur, and Alejandro Mota. The Schwarz alternating method for the seamless coupling of nonlinear reduced order models and full order models. *arXiv preprint arXiv:2210.12551*, 2022.

[294] Ricardo Reyes and Ramon Codina. Element boundary terms in reduced order models for flow problems: Domain decomposition and adaptive coarse mesh hyper-reduction. *Computer Methods in Applied Mechanics and Engineering*, 368:113159, 2020.

[295] Alejandro Mota, Daria Koliesnikova, Irina Tezaur, and Jonathan Hoy. Fundamentally new coupled approach to contact mechanics via the Dirichlet-Neumann Schwarz alternating method. *arXiv preprint arXiv:2311.05643*, 2023.

[296] Daria Koliesnikova, Alejandro Mota, and Irina Kalashnikova Tezaur. 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), 2023.

[297] Elena Zappon, Andrea Manzoni, Paola Gervasio, and Alfio Quarteroni. A reduced order model for domain decompositions with non-conforming interfaces. *Journal of Scientific Computing*, 99(1):22, 2024.

[298] Jan Vierendeels, Lieve Lanoye, Joris Degroote, and Pascal Verdonck. Implicit coupling of partitioned fluid-structure interaction problems with reduced order models. *Computers & Structures*, 85(11-14):970–976, June 2007. ISSN 0045-7949. doi:

10.1016/j.compstruc.2006.11.006. URL https://linkinghub.elsevier.com/retrieve/pii/S0045794906003865. Publisher: Elsevier BV.

[299] Matteo Aletti and Damiano Lombardi. A reduced-order representation of the Poincaré–Steklov operator: an application to coupled multi-physics problems. *International Journal for Numerical Methods in Engineering*, 111(6):581–600, 2017.

[300] Monica Nonino, Francesco Ballarin, Gianluigi Rozza, and Yvon Maday. Projection based semi-implicit partitioned reduced basis method for fluid-structure interaction problems. *Journal of Scientific Computing*, 94(1), January 2023. ISSN 0885-7474, 1573-7691. doi: 10.1007/s10915-022-02049-6. URL https://link.springer.com/10.1007/s10915-022-02049-6. Publisher: Springer Science and Business Media LLC.

[301] Francesco Ballarin, Sanghyun Lee, and Son-Young Yi. Projection-based reduced order modeling of an iterative scheme for linear thermo-poroelasticity. *Results in Applied Mathematics*, 21:100430, February 2024. ISSN 25900374. doi: 10.1016/j.rinam.2023.100430. URL https://linkinghub.elsevier.com/retrieve/pii/S2590037423000766.

[302] Shi Chen, Zhiyan Ding, Qin Li, and Stephen J Wright. A reduced order Schwarz method for nonlinear multiscale elliptic equations based on two-layer neural networks. *arXiv preprint arXiv:2111.02280*, 2021.

[303] D. Xiao, P. Yang, F. Fang, J. Xiang, C.C. Pain, and I.M. Navon. Non-intrusive reduced order modelling of fluid-structure interactions. *Computer Methods in Applied Mechanics and Engineering*, 303:35–54, May 2016. ISSN 0045-7825. doi: 10.1016/j.cma.2015.12.029. URL https://linkinghub.elsevier.com/retrieve/pii/S0045782516300068. Publisher: Elsevier BV.

[304] Xinshuai Zhang, Tingwei Ji, Fangfang Xie, Changdong Zheng, and Yao Zheng. Data-driven nonlinear reduced-order modeling of unsteady fluid-structure interactions. *Physics of Fluids*, 34(5), May 2022. ISSN 1070-6631, 1089-7666. doi: 10.1063/5.0090394. URL https://pubs.aip.org/pof/article/34/5/053608/2847123/Data-driven-nonlinear-reduced-order-modeling-of. Publisher: AIP Publishing.

[305] Renkun Han, Yixing Wang, Weiqi Qian, Wenzheng Wang, Miao Zhang, and Gang Chen. Deep neural network based reduced-order model for fluid-structure interaction system. *Physics of Fluids*, 34(7), July 2022. ISSN 1070-6631, 1089-7666. doi: 10.1063/5.0096432. URL https://pubs.aip.org/pof/article/34/7/073610/2846978/Deep-neural-network-based-reduced-order-model-for. Publisher: AIP Publishing.

[306] M. Barzegar Gerdroodbary and Sajad Salavatidezfouli. A predictive surrogate model based on linear and nonlinear solution manifold reduction in cardiovascular FSI: A comparative study. *Computers in Biology and Medicine*, 189:109959, May 2025. ISSN 0010-4825. doi: 10.1016/j.compbiomed.2025.109959. URL https://linkinghub.elsevier.com/retrieve/pii/S0010482525003105. Publisher: Elsevier BV.

[307] Giorgio Gobat, Stefania Fresca, Andrea Manzoni, and Attilio Frangi. Reduced Order Modeling of nonlinear vibrating multiphysics microstructures with deep learning-based approaches. *Sensors*, 23(6):3001, March 2023. ISSN 1424-8220. doi: 10.3390/s23063001. URL `https://www.mdpi.com/1424-8220/23/6/3001`. Publisher: MDPI AG.

[308] D Xiao, CE Heaney, F Fang, L Mottet, R Hu, DA Bistrian, E Aristodemou, IM Navon, and CC Pain. A domain decomposition non-intrusive reduced order model for turbulent flows. *Computers & Fluids*, 182:15–27, 2019.

[309] Cong Xiao, Olwijn Leeuwenburgh, Hai Xiang Lin, and Arnold Heemink. Non-intrusive subdomain POD-TPWL for reservoir history matching. *Computational Geosciences*, 23: 537–565, 2019.

[310] Cong Xiao, Olwijn Leeuwenburgh, Hai Xiang Lin, and Arnold Heemink. Subdomain POD-TPWL with local parameterization for large-scale reservoir history matching problems. *arXiv preprint arXiv:1901.08059*, 2019.

[311] Cong Xiao, Olwijn Leeuwenburgh, Hai Xiang Lin, and Arnold Heemink. Efficient estimation of space varying parameters in numerical models using non-intrusive subdomain reduced order modeling. *Journal of Computational Physics*, 424:109867, 2021.

[312] Rossella Arcucci, César Quilodrán Casas, Dunhui Xiao, Laetitia Mottet, Fangxin Fang, Pin Wu, Christopher Pain, and Yi-Ke Guo. A domain decomposition reduced order model with data assimilation (DD-RODA). In *Parallel Computing: Technology Trends*, pages 189–198. IOS Press, 2020.

[313] Rossella Arcucci, Laetitia Mottet, César A Quilodrán Casas, Florian Guitton, Christopher Pain, and Yi-Ke Guo. Adaptive domain decomposition for effective data assimilation. In *Euro-Par 2019: Parallel Processing Workshops: Euro-Par 2019 International Workshops, Göttingen, Germany, August 26–30, 2019, Revised Selected Papers 25*, pages 583–595. Springer, 2020.

[314] Alejandro N Diaz, Youngsoo Choi, and Matthias Heinkenschloss. Nonlinear-manifold reduced order models with domain decomposition. *arXiv preprint arXiv:2312.00713*, 2023.

[315] Alejandro N Diaz, Youngsoo Choi, and Matthias Heinkenschloss. A fast and accurate domain decomposition nonlinear manifold reduced order model. *Computer Methods in Applied Mechanics and Engineering*, 425:116943, 2024.

[316] Nikhil Iyengar, Dushhyanth Rajaram, Kenneth Decker, Christian Perron, and Dimitri N Mavris. Nonlinear reduced order modeling using domain decomposition. In *AIAA SciTech 2022 Forum*, page 1250, 2022.

[317] Nikhil Iyengar, Dushhyanth Rajaram, and Dimitri Mavris. Domain decomposition strategy for combining nonlinear and linear reduced-order models. *AIAA Journal*, 62(4):1375–1389, 2024.

[318] Ameya D Jagtap, Ehsan Kharazmi, and George Em Karniadakis. Conservative physics-informed neural networks on discrete domains for conservation laws: Applications to forward and inverse problems. *Computer Methods in Applied Mechanics and Engineering*, 365:113028, 2020.

[319] Ameya D Jagtap and George Em Karniadakis. Extended physics-informed neural networks (XPINNs): A generalized space-time domain decomposition based deep learning framework for nonlinear partial differential equations. *Communications in Computational Physics*, 28(5), 2020.

[320] Khemraj Shukla, Ameya D Jagtap, and George Em Karniadakis. Parallel physics-informed neural networks via domain decomposition. *Journal of Computational Physics*, 447:110683, 2021.

[321] Ehsan Kharazmi, Zhongqiang Zhang, and George Em Karniadakis. hp-VPINNs: Variational physics-informed neural networks with domain decomposition. *Computer Methods in Applied Mechanics and Engineering*, 374:113547, 2021.

[322] Vikas Dwivedi, Nishant Parashar, and Balaji Srinivasan. Distributed learning machines for solving forward and inverse problems in partial differential equations. *Neurocomputing*, 420:299–316, 2021.

[323] Zheyuan Hu, Ameya D Jagtap, George Em Karniadakis, and Kenji Kawaguchi. When do extended physics-informed neural networks (XPINNs) improve generalization? *SIAM Journal on Scientific Computing*, 44(5):A3158–A3182, 2022.

[324] Xinyu Pan and Dunhui Xiao. Domain decomposition for physics-data combined neural network based parametric reduced order modelling. *Journal of Computational Physics*, 519: 113452, 2024.

[325] Ke Li, Kejun Tang, Tianfan Wu, and Qifeng Liao. D3M: A deep domain decomposition method for partial differential equations. *Ieee Access*, 8:5283–5294, 2019.

[326] Wuyang Li, Xueshuang Xiang, and Yingxiang Xu. Deep domain decomposition method: Elliptic problems. In *Mathematical and Scientific Machine Learning*, pages 269–286. PMLR, 2020.

[327] Ben Moseley, Andrew Markham, and Tarje Nissen-Meyer. Finite basis physics-informed neural networks (FBPINNs): a scalable domain decomposition approach for solving differential equations. *Advances in Computational Mathematics*, 49(4):62, 2023.

[328] Victorita Dolean, Alexander Heinlein, Siddhartha Mishra, and Ben Moseley. Finite basis physics-informed neural networks as a Schwarz domain decomposition method. In *International Conference on Domain Decomposition Methods*, pages 165–172. Springer, 2022.

[329] Daria Hrebenshchykova. Multilevel and distributed physics-informed neural networks for the helmholtz equation. 2023.

[330] Victorita Dolean, Alexander Heinlein, Siddhartha Mishra, and Ben Moseley. Multilevel domain decomposition-based architectures for physics-informed neural networks. *Computer Methods in Applied Mechanics and Engineering*, 429:117116, 2024.

[331] Will Snyder, Irina Tezaur, and Christopher Wentland. Domain decomposition-based coupling of physics-informed neural networks via the Schwarz alternating method. *arXiv preprint arXiv:2311.00224*, 2023.

[332] SiHun Lee, Kijoo Jang, Sangmin Lee, Haeseong Cho, and SangJoon Shin. Parametric model order reduction by machine learning for fluid-structure interaction analysis. *Engineering with Computers*, 40(1):45–60, February 2024. ISSN 0177-0667, 1435-5663. doi: 10.1007/s00366-023-01782-2. URL https://link.springer.com/10.1007/s00366-023-01782-2. Publisher: Springer Science and Business Media LLC.

[333] Afrah Farea, Saiful Khan, Reza Daryani, Emre Cenk Ersan, and Mustafa Serdar Celebi. Learning fluid-structure interaction dynamics with physics-informed neural networks and immersed boundary methods, June 2025. URL http://arxiv.org/abs/2505.18565. arXiv:2505.18565 [cs].

[334] Zhiyong Wu, Huan Wang, Chang He, Bingjian Zhang, Tao Xu, and Qinglin Chen. The application of physics-informed machine learning in multiphysics modeling in chemical engineering. *Industrial & Engineering Chemistry Research*, 62(44):18178–18204, November 2023. ISSN 0888-5885, 1520-5045. doi: 10.1021/acs.iecr.3c02383. URL https://pubs.acs.org/doi/10.1021/acs.iecr.3c02383. Publisher: American Chemical Society (ACS).

[335] Yubin Ryu, Sunkyu Shin, Won Bo Lee, and Jonggeol Na. Multiphysics generalization in a polymerization reactor using physics-informed neural networks. *Chemical Engineering Science*, 298:120385, October 2024. ISSN 0009-2509. doi: 10.1016/j.ces.2024.120385. URL https://linkinghub.elsevier.com/retrieve/pii/S0009250924006857. Publisher: Elsevier BV.

[336] Yaoyao Ma, Xiaoyu Xu, Shuai Yan, and Zhuoxiang Ren. A preliminary study on the resolution of electro-thermal multi-physics coupling problem using physics-informed neural network (PINN). *Algorithms*, 15(2):53, February 2022. ISSN 1999-4893. doi: 10.3390/a15020053. URL https://www.mdpi.com/1999-4893/15/2/53. Publisher: MDPI AG.

[337] Christopher Greenshields and Henry Weller. *Notes on Computational Fluid Dynamics: General Principles*. CFD Direct Ltd, Reading, UK, 2022.

[338] Osborne Reynolds. IV. On the dynamical theory of incompressible viscous fluids and the determination of the criterion. *Philosophical Transactions of the Royal Society of London.*

(A.), 186:123–164, December 1895. ISSN 0264-3820, 2053-9231. doi: 10.1098/rsta.1895.0004.

[339] Joseph Boussinesq. *Essai Sur La Théorie Des Eaux Courantes*. Impr. nationale.

[340] L. Prandtl. Bericht über Untersuchungen zur ausgebildeten Turbulenz. *ZAMM - Journal of Applied Mathematics and Mechanics / Zeitschrift für Angewandte Mathematik und Mechanik*, 5(2):136–139, 1925. ISSN 1521-4001. doi: 10.1002/zamm.19250050212. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/zamm.19250050212. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/zamm.19250050212.

[341] P. Spalart and S. Allmaras. A one-equation turbulence model for aerodynamic flows. In *30th Aerospace Sciences Meeting and Exhibit*, Reno,NV,U.S.A., January 1992. American Institute of Aeronautics and Astronautics. doi: 10.2514/6.1992-439. URL https://arc.aiaa.org/doi/10.2514/6.1992-439.

[342] B.E. Launder and D.B. Spalding. The numerical computation of turbulent flows. *Computer Methods in Applied Mechanics and Engineering*, 3(2):269–289, March 1974. ISSN 0045-7825. doi: 10.1016/0045-7825(74)90029-2. URL https://linkinghub.elsevier.com/retrieve/pii/0045782574900292. Publisher: Elsevier BV.

[343] David C. Wilcox. Dilatation-dissipation corrections for advanced turbulence models. *AIAA Journal*, 30(11):2639–2646, November 1992. ISSN 0001-1452, 1533-385X. doi: 10.2514/3.11279. URL https://arc.aiaa.org/doi/10.2514/3.11279. Publisher: American Institute of Aeronautics and Astronautics (AIAA).

[344] Florian R Menter. Improved two-equation k-ω turbulence models for aerodynamic flows. Technical report, 1992.

[345] Brian Launder, G. Reece, and W. Rodi. Progress in the development of a Reynolds stress turbulence closure. *Journal of Fluid Mechanics*, 68:537–566, April 1975. doi: 10.1017/S0022112075001814.

[346] F. R. Menter. Two-equation eddy-viscosity turbulence models for engineering applications. *AIAA Journal*, 32(8):1598–1605, August 1994. ISSN 0001-1452, 1533-385X. doi: 10.2514/3.12149. URL https://arc.aiaa.org/doi/10.2514/3.12149. Publisher: American Institute of Aeronautics and Astronautics (AIAA).

[347] Florian Menter, M. Kuntz, and RB Langtry. Ten years of industrial experience with the SST turbulence model. *Heat and Mass Transfer*, 4, January 2003.

[348] Clark Pederson and Ethan Vogel. Menter Shear Stress Transport Model, 10-04-2024. URL https://turbmodels.larc.nasa.gov/sst.html.

[349] Hrvoje Jasak, Aleksandar Jemcov, Zeljko Tukovic, et al. OpenFOAM: A C++ library for complex physics simulations. In *International workshop on coupled methods in numerical dynamics*, volume 1000, pages 1–20. Dubrovnik, Croatia), 2007.

[350] Hrvoje Jasak. OpenFOAM: Open source CFD in research and industry. *International journal of naval architecture and ocean engineering*, 1(2):89–94, 2009.

[351] OpenCFD Ltd. *OpenFOAM®: Open source CFD : Documentation*, 2024. URL `https://doc.openfoam.com/2312/`.

[352] Suhas V Patankar. A calculation procedure for two-dimensional elliptic situations. *Numerical heat transfer*, 4(4):409–425, 1981.

[353] Suhas V Patankar and D Brian Spalding. A calculation procedure for heat, mass and momentum transfer in three-dimensional parabolic flows. In *Numerical prediction of flow, heat transfer, turbulence and combustion*, pages 54–73. Elsevier, 1983.

[354] Steven L Brunton and J Nathan Kutz. *Data-driven science and engineering: Machine learning, dynamical systems, and control*. Cambridge University Press, 2024.

[355] Erhard Schmidt. Zur Theorie der linearen und nichtlinearen Integralgleichungen. *Mathematische Annalen*, 63(4):433–476, 1907. URL `https://scholar.archive.org/work/73zcd4f37jdevl5xrzbk2iviai/access/ia_file/crossref-pre-1909-scholarly-works/10.1007%252Fbf01446680.zip/10.1007%252Fbf01449770.pdf`. Publisher: Springer Berlin/Heidelberg.

[356] Carl Eckart and Gale Young. The approximation of one matrix by another of lower rank. *Psychometrika*, 1(3):211–218, 1936. URL `https://www.cambridge.org/core/journals/psychometrika/article/approximation-of-one-matrix-by-another-of-lower-rank/B29672E1EDD0FA1B7611D4DFAFC321B3`. Publisher: Springer-Verlag.

[357] Max D. Gunzburger, Janet S. Peterson, and John N. Shadid. Reduced-order modeling of time-dependent PDEs with multiple parameters in the boundary data. 196(4):1030–1047. ISSN 0045-7825. doi: 10.1016/j.cma.2006.08.004. URL `https://www.sciencedirect.com/science/article/pii/S0045782506002337`.

[358] Pierfrancesco Siena, Pasquale Claudio Africa, Michele Girfoglio, and Gianluigi Rozza. A hybrid Reduced Order Model to enforce outflow pressure boundary conditions in computational haemodynamics, January 2025.

[359] Lorenzo Vergari, Antonio Cammi, and Stefano Lorenzi. Reduced order modeling approach for parametrized thermal-hydraulics problems: Inclusion of the energy equation in the POD-FV-ROM method. 118:103071. ISSN 0149-1970. doi: 10.1016/j.pnucene.2019.103071. URL `https://www.sciencedirect.com/science/article/pii/S0149197019301660`.

[360] Jorge Yanez and Andreas G. Class. Reduced Order Model of standard $k - \varepsilon$ turbulence model. *Computers & Fluids*, 245:105608, September 2022. ISSN 0045-7930. doi: 10.1016/j.compfluid.2022.105608. URL https://www.sciencedirect.com/science/article/pii/S0045793022002134.

[361] Jorge Yanez and Andreas G. Class. Uncertainty quantification of a spent fuel refrigeration pool utilizing a $k-\varepsilon$ turbulence reduced order model. *Nuclear Engineering and Design*, 412:112436, October 2023. ISSN 0029-5493. doi: 10.1016/j.nucengdes.2023.112436. URL https://www.sciencedirect.com/science/article/pii/S0029549323002856.

[362] Hans Johnston and Jian-Guo Liu. Accurate, stable and efficient Navier-Stokes solvers based on explicit treatment of the pressure term. *Journal of Computational Physics*, 199(1):221–259, September 2004. ISSN 0021-9991. doi: 10.1016/j.jcp.2004.02.009. URL https://www.sciencedirect.com/science/article/pii/S002199910400083X.

[363] Alfonso Caiazzo, Traian Iliescu, Volker John, and Swetlana Schyschlowa. A numerical investigation of velocity–pressure reduced order models for incompressible flows. *Journal of Computational Physics*, 259:598–616, February 2014. ISSN 00219991. doi: 10.1016/j.jcp.2013.12.004.

[364] Andrea Manzoni, Alfio Quarteroni, and Gianluigi Rozza. Shape optimization for viscous flows by reduced basis methods and free-form deformation. *International Journal for Numerical Methods in Fluids*, 70(5):646–670, October 2012. ISSN 0271-2091, 1097-0363. doi: 10.1002/fld.2712. URL https://onlinelibrary.wiley.com/doi/10.1002/fld.2712.

[365] Gianluigi Rozza, Andrea Manzoni, and Federico Negri. Reduction Strategies for Pde-Constrained Optimization Problems in Haemodynamics. Vienna, Austria, September 2012.

[366] Jens L Eftang and Anthony T Patera. A port-reduced static condensation reduced basis element method for large component-synthesized structures: approximation and a posteriori error estimation. *Advanced Modeling and Simulation in Engineering Sciences*, 1:1–49, 2014.

[367] Federico Pichi, Maria Strazzullo, Francesco Ballarin, and Gianluigi Rozza. Driving bifurcating parametrized nonlinear PDEs by optimal control strategies: application to Navier-Stokes equations with model order reduction. *ESAIM: Mathematical Modelling and Numerical Analysis*, 56(4):1361–1400, July 2022. ISSN 2822-7840, 2804-7214. doi: 10.1051/m2an/2022044. URL https://www.esaim-m2an.org/10.1051/m2an/2022044.

[368] Giovanni Stabile and Gianluigi Rozza. Finite volume POD-Galerkin stabilised reduced order methods for the parametrised incompressible Navier-Stokes equations. *Computers & Fluids*, 173:273–284, September 2018. ISSN 00457930. doi: 10.1016/j.compfluid.2018.01.035.

[369] Charles R. Harris, K. Jarrod Millman, Stéfan J. van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fernández del Río, Mark Wiebe, Pearu Peterson, Pierre Gérard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E. Oliphant. Array programming with NumPy. *Nature*, 585(7825):357–362, September 2020. doi: 10.1038/s41586-020-2649-2. URL `https://doi.org/10.1038/s41586-020-2649-2`.

[370] Pauli Virtanen, Ralf Gommers, Evgeni Burovski, Travis E Oliphant, Warren Weckesser, David Cournapeau, Pearu Peterson, Tyler Reddy, Matt Haberland, Josh Wilson, et al. scipy/scipy: SciPy 1.6. 0. *Zenodo*, 2021.

[371] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[372] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. PyTorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

[373] Martin D. Buhmann. *Radial Basis Functions: Theory and Implementations*. Cambridge Monographs on Applied and Computational Mathematics. Cambridge University Press, 2003.

[374] Robert Schaback. A practical guide to radial basis functions. *Electronic Resource*, 11: 1–12, 2007.

[375] Christopher KI Williams and Carl Edward Rasmussen. *Gaussian processes for machine learning*, volume 2. MIT press Cambridge, MA, 2006.

[376] Jie Wang. An intuitive tutorial to Gaussian processes regression. *Computing in Science & Engineering*, 2023.

[377] Christopher M. Bishop and Nasser M. Nasrabadi. *Pattern Recognition and Machine Learning*, volume 4. Springer, 2006.

[378] KL Priddy. *Artificial Neural Networks: an Introduction*. Prentice-Hall, India, 2005.

[379] Tomasz Szandała. Review and comparison of commonly used activation functions for deep neural networks. In *Bio-inspired neurocomputing*, pages 203–224. Springer, 2020.

[380] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, January 2017.

[381] Francois Chollet. *Deep learning with Python*. simon and schuster, 2021.

[382] J. Pacio, M. Daubner, F. Fellmoser, K. Litfin, and Th. Wetzel. Experimental study of heavy-liquid metal (LBE) flow and heat transfer along a hexagonal 19-rod bundle with wire spacers. *Nuclear Engineering and Design*, 301:111–127, May 2016. ISSN 00295493. doi: 10.1016/j.nucengdes.2016.03.003.

[383] J. Pacio, M. Daubner, F. Fellmoser, K. Litfin, and T. Wetzel. Heat transfer experiment in a partially (internally) blocked 19-rod bundle with wire spacers cooled by LBE. *Nuclear Engineering and Design*, 330:225–240, April 2018. ISSN 00295493. doi: 10.1016/j.nucengdes.2018.01.034.

[384] Bruce R. Munson, Theodore H. Okiishi, Wade W. Huebsch, and Alric P. Rothmayer. *Fundamentals of Fluid Mechanics*. John Wiley & Sons, Inc., 2013.

[385] Yan Zhang, Chenglong Wang, Rong Cai, Zhike Lan, Yaou Shen, Dalin Zhang, Wenxi Tian, Guanghui Su, and Suizheng Qiu. Experimental investigation on flow and heat transfer characteristics of lead–bismuth eutectic in circular tubes. *Applied Thermal Engineering*, 180: 115820, November 2020. ISSN 13594311. doi: 10.1016/j.applthermaleng.2020.115820.

[386] Charles Borden Johnson. *Power-Law Velocity-Profile-Exponent Variations with Reynolds Number, Wall Cooling, and Mach Number in a Turbulent Boundary Layer*. National Aeronautics and Space Administration, 1969.

[387] Gaël Guennebaud, Benoît Jacob, et al. Eigen v3. http://eigen.tuxfamily.org, 2010.

[388] SG Forbes, TR Wilson, and GF Brockett. *Reactor Systems Response Analytical Development Program*. Atomic Energy Division, Idaho Operations Office, US Atomic Energy Commission, 1969.

[389] Mohammad Pourgol-Mohammad. Thermal–hydraulics system codes uncertainty assessment: A review of the methodologies. *Annals of Nuclear Energy*, 36(11-12):1774–1786, 2009.

[390] Jorge Sanchez-Torrijos, Pau Aragón, Cesar Queral, Francisco Feria, Arndt Schubert, and Paul Van Uffelen. A review on the thermo-mechanical modelling needs in system codes. *Nuclear Engineering and Design*, 406:112243, 2023.

[391] Z Cheng. Strategies for developing subchannel capability in an advanced system thermal-hydraulic code: A literature review. 2014.

[392] Haiqi Zhao, Daogang Lu, Yuhao Zhang, and Xueyuan Zhang. Numerical simulation of natural circulation characteristics under different DHX layout schemes in pool-type SFR during station blackout accident. *Progress in Nuclear Energy*, 157:104567, 2023.

[393] Haiqi Zhao, Yuhao Zhang, Qifeng Pu, Haijie Song, Danting Sui, Lu Liu, and Daogang Lu. Analysis on thermal hydraulic characteristics for PHENIX natural circulation system using "1D system code+ 3D cfd" simulation method. *Progress in Nuclear Energy*, 176:105370, 2024.

[394] Marc-Olivier G Delchini, Emilian L Popov, William David Pointer, and Laura P Swiler. Assessment of SFR wire wrap simulation uncertainties. Technical report, Oak Ridge National Lab.(ORNL), Oak Ridge, TN (United States), 2016.

[395] Heleen Doolaard, Afaque Shams, Ferry Roelofs, Katrien Van Tichelen, Steven Keijers, Jeroen De Ridder, Joris Degroote, Jan Vierendeels, Ivan Di Piazza, R Marinari, et al. CFD benchmark for a heavy liquid metal fuel assembly. In *16th International Topical Meeting on Nuclear Reactor Thermal Hydraulics (NURETH-16)*, pages 2196–2208. American Nuclear Society (ANS), 2015.

[396] Mariano Tarantino, Ferry Roelofs, Afaque Shams, Abdalla Batta, Vincent Moreau, Ivan Di Piazza, Antoine Gershenfeld, and Philippe Planquart. SESAME project: advancements in liquid metal thermal hydraulics experiments and simulations. *EPJ N-Nuclear Sciences & Technologies*, 6:18, 2020.

[397] Rongjie Li, Dajun Fan, Minghan He, Ruoxiang Qiu, Yanze Tang, Wangsheng Tian, and Long Gu. PIV and CFD study on crossflow characteristics in a 7-pin wire-wrapped bundle channel. *International Journal of Energy Research*, 2023(1):5542001, 2023.

[398] Xiang Chai, Xiaojing Liu, Jinbiao Xiong, and Xu Cheng. CFD analysis of flow blockage phenomena in a LBE-cooled 19-pin wire-wrapped rod bundle. *Nuclear Engineering and Design*, 344:107–121, 2019.

[399] Aleksandr V Obabko, Elia Merzari, and Paul F Fischer. Nek5000 large-eddy simulations for thermal-hydraulics of deformed wire-wrap fuel assemblies. *Transactions of the American Nuclear Society*, 115, 2016.

[400] Elia Merzari, Paul Fischer, Misun Min, Stefan Kerkemeier, Aleksandr Obabko, Dillon Shaver, Haomin Yuan, Yiqi Yu, Javier Martinez, Landon Brockmeyer, et al. Toward exascale: overview of large eddy simulations and direct numerical simulations of nuclear reactor flows with the spectral element method in Nek5000. *Nuclear Technology*, 206(9):1308–1324, 2020.

[401] Maciej Majchrzak, Katarzyna Marciniak-Lukasiak, and Piotr Lukasiak. A survey on the application of machine learning in turbulent flow simulations. *Energies*, 16(4):1755, 2023.

[402] Law of the wall, May 2025. URL `https://en.wikipedia.org/w/index.php?title=Law_of_the_wall&oldid=1292086251`. Page Version ID: 1292086251.

[403] Robert F Warming and Richard M Beam. Upwind second-order difference schemes and applications in aerodynamic flows. *AIAA Journal*, 14(9):1241–1249, 1976.

[404] Mohamed Aissa. *GPU-accelerated CFD Simulations for Turbomachinery Design Optimization*. PhD thesis, Delft University of Technology, October 2017.

[405] Joel H Ferziger, Milovan Perić, and Robert L Street. *Computational methods for fluid dynamics*. springer, 2019.