

RESEARCH ARTICLE

A brief review of reduced order models using intrusive and non-intrusive techniques

Guglielmo Padula | Michele Girfoglio | Gianlugi Rozza 

MathLab, Mathematics Area, SISSA International School for Advanced Studies, Trieste, Italy

Correspondence

Gianlugi Rozza, MathLab, Mathematics Area, SISSA International School for Advanced Studies, Trieste, Italy.
Email: grozza@sissa.it

Abstract

Reduced Order Models (ROMs) have gained a great attention by the scientific community in the last years thanks to their capabilities of significantly reducing the computational cost of the numerical simulations, which is a crucial objective in applications like real time control and shape optimization. This contribution aims to provide a brief overview about such a topic. We discuss both a classic intrusive framework based on a Galerkin projection technique and hybrid/non-intrusive approaches, including Physics Informed Neural Networks (PINN), purely Data-Driven Neural Networks (NN), Radial Basis Functions (RBF), Dynamic Mode Decomposition (DMD) and Gaussian Process Regression (GPR). We also briefly mention geometrical parametrization and dimensionality reduction methods like Active Subspaces (ASs). Then we test the performance of such approaches in terms of efficiency and accuracy against three academic test cases, the lid driven cavity, the flow past a cylinder and the geometrically parametrized Stanford Bunny. Moreover, we also briefly present some preliminary results related to a more complex case involving an industrial application.

1 | INTRODUCTION

The increase in computing resources over the recent decades has continuously fostered the development of numerical methods and enabled the utilization of complex algorithms. However, in certain cases, including real time control of systems like digital twins [1, 2] or optimization problems [3, 4], it is still very hard to obtain an high computational efficiency.

There are mainly two ways to solve this problem. The first one is to use High Performance Computing (HPC) clusters [5] but they are immobile, costly to build and are large electrical consumers. The second one is to introduce Reduced Order Models (ROMs) enabling fast computation even on small computing facilities, though with a loss in accuracy.

The development of a ROM is typically split in two stages:

- The *offline stage*, in which for few selected values of the parameters (physical or geometrical) involved in the problem, high-fidelity simulations are computed, typically by using an HPC facility. Such simulations, obtained by solving

This is an open access article under the terms of the [Creative Commons Attribution-NonCommercial-NoDerivs](https://creativecommons.org/licenses/by-nc-nd/4.0/) License, which permits use and distribution in any medium, provided the original work is properly cited, the use is non-commercial and no modifications or adaptations are made.

© 2024 The Author(s). Proceedings in Applied Mathematics and Mechanics published by Wiley-VCH GmbH.

the discretized governing equations system, hereinafter referred to as Full Order Model (FOM), are computationally expensive, but they are performed only once. At this point one makes use of some techniques, among which Proper Orthogonal Decomposition (POD) [6, 7], to extract the essential information from the high-fidelity database and to assemble the surrogate model.

- The *online stage*, in which, taking advantage of the surrogate model, the solution is efficiently predicted for new values of the parameters. The computational speedup with respect to the FOM could be significantly high: for instance, for some fluid dynamics problems, a value of 10^5 is achievable (see Section 3).

ROMs are divided into two big families [7]: intrusive methods, in which one manipulates directly the governing equations, and non-intrusive methods, in which only the simulation data are considered. It should be noted that also for non-intrusive methods one could make use of information coming from the physics of the system at hand (see Section 2).

The paper is organized as follows. In Section 2, a classic intrusive technique, the Galerkin projection method [6], and several data-driven techniques, Physics Informed Neural Networks (PINN) [1, 7], Radial Basis Functions (RBF) [8–15], purely Data-Driven Neural Network (NN) [7, 9, 11, 14–23], Dynamic Mode Decomposition (DMD) [7, 12, 19, 24] and Gaussian Process Regression (GPR) [7, 9, 10, 14] are briefly discussed. An example of geometrical parametrization is also introduced and the Active Subspace (AS) method is presented as a possible solution strategy for dimensionality reduction. In Section 3 we present some numerical results related to three academic test cases and in Section 4 we provide some insights about an industrial application. Finally in Section 5 we report some concluding remarks.

2 | REDUCED ORDER MODELS

We refer to ROMs to accelerate the numerical solution of parameterized Partial Differential Equations (PDEs) in the space-time domain, which represents our FOM and whose general form is the following:

$$\begin{cases} \mathcal{L}(u, \mu, t, x, y, z) = 0 & \text{with } (x, y, z) \in \Omega(\mu), \quad t \in (0, T], \quad \mu \in \mathbb{P}, \\ \mathcal{B}(u, \mu, t, x, y, z) = 0 & \text{with } (x, y, z) \in \partial\Omega(\mu), \quad t \in (0, T], \quad \mu \in \mathbb{P}, \\ \mathcal{T}(u, \mu, x, y, z) = 0 & \text{with } (x, y, z) \in \Omega(\mu), \quad t = 0, \quad \mu \in \mathbb{P}, \end{cases} \quad (1)$$

where μ is a parameter (physical or geometrical), $\Omega(\mu) \subset \mathbb{R}^3$ is the space domain of the problem, T is the final time of the simulation, \mathbb{P} is the parameter space, $u \in \mathbb{V}$ and $\mathbb{V} \subseteq \{v | v : \mathbb{P} \times \mathbb{R}^3 \rightarrow \mathbb{R}^n\}$ is an Hilbert space. Finally $\mathcal{L} : \mathbb{V} \times \mathbb{P} \times \mathbb{R}^+ \times \mathbb{R}^3 \rightarrow \mathbb{R}$, $\mathcal{B} : \mathbb{V} \times \mathbb{P} \times \mathbb{R}^+ \times \mathbb{R}^3 \rightarrow \mathbb{R}$, $\mathcal{T} : \mathbb{V} \times \mathbb{P} \times \mathbb{R}^3 \rightarrow \mathbb{R}$ are continuous functions whose explicit formulation depends on the problem at hand.

2.1 | An intrusive method: The Galerkin projection

For sake of simplicity, we assume that:

- the problem is steady, so \mathcal{L} , \mathcal{B} and \mathcal{T} do not depend on t .
- Ω does not depend on μ .
- $\mathbb{V} = \{v \in \mathbb{P} \times \mathbb{R}^3 | v(\mu, \cdot) \in H^1(\mathbb{R}^3) \text{ and } v(\mu, \cdot)_{\partial\Omega} = 0 \forall \mu \in \mathbb{P}\}$, so in this case $n = 1$.
- Based on the affine assumption, the first equation of system (1) can be rewritten as

$$\sum_{i=1}^m \theta_{\mathcal{L}}^i(\mu) \mathcal{L}_i(u, x, y, z) = \sum_{i=1}^m \theta_f^i(\mu) f_i(x, y, z) \quad \forall (x, y, z) \in \Omega, \mu \in \mathbb{P}, \quad (2)$$

where $\theta_{\mathcal{L}}^i$ and θ_f^i are scalar continuous functions from \mathbb{P} to \mathbb{R} , f_i are continuous functions from \mathbb{R}^3 to \mathbb{R} and \mathcal{L}_i are continuous functions from $\mathbb{V} \times \mathbb{R}^3$ to \mathbb{R} .

- The second equation of system (1) reads

$$u(\mu, x, y, z) = 0 \quad \forall (x, y, z) \in \partial\Omega, \mu \in \mathbb{P}. \quad (3)$$

By multiplying Equation (2) by $v \in \mathbb{W} = \{v : \mathbb{R}^3 \rightarrow \mathbb{R} | v|_{\partial\Omega} = 0\}$ and integrating over Ω we obtain the weak form of the problem:

$$\sum_{i=1}^m \theta_{\mathcal{L}}^i(\mu) \mathcal{A}_i(u(\mu, \cdot), v) = \sum_{i=1}^m \theta_f^i(\mu) \mathcal{F}_i(v) \quad \forall v \in \mathbb{W}, \mu \in \mathbb{P}, \quad (4)$$

where

$$\mathcal{A}_i(u(\mu, \cdot), v) = \int_{\Omega} u(\mu, x, y, z) v(x, y, z) dx dy dz \quad \text{and} \quad \mathcal{F}_i(v) = \int_{\Omega} f_i(x, y, z) v(x, y, z) dx dy dz. \quad (5)$$

We assume that \mathcal{A}_i are bilinear with respect to u, v . Moreover, we also assume that they are symmetric and coercive, that is,

$$\mathcal{A}_i(u, v) > c_i \|u\|_W \|v\|_W \quad c_i > 0 \quad \forall u, v \in \mathbb{W}, \forall i = 1, 2, \dots, m. \quad (6)$$

In this setting, it can be proven that a solution $u \in \mathbb{V}$ exists and is unique [25]. As the \mathcal{A}_i are bilinear we can build the space $\mathbb{W}_N = \text{span}\{v_1, \dots, v_N\}$. We can restrict to $u \in \{u \in \mathbb{V} | u(\mu, \cdot) \in \mathbb{W}_N, \forall \mu \in \mathbb{P}\}$, so it follows that

$$u = \sum_{j=1}^N \alpha_j(\mu) v_j, \quad (7)$$

where $\alpha_j : \mathbb{P} \rightarrow \mathbb{R}$. By substituting Equation (7) in Equation (4) we obtain the following system of N equations

$$\sum_{i=1}^m \sum_{j=1}^N \alpha_j(\mu) \theta_{\mathcal{L}}^i(\mu) \mathcal{A}_i(v_j, v_k) = \sum_{i=1}^m \theta_f^i(\mu) \mathcal{F}_i(v_k) \quad k = 1, \dots, N. \quad (8)$$

Note that once a basis of \mathbb{W}_N has been chosen, the computation of the \mathcal{A}_i and \mathcal{F}_i (which can be very costly) is done only once. We assume that it does exist a reduced space \mathbb{W}_{rb} of dimension $N_{rb} \ll N$ such that the error

$$\varepsilon = \|u(\mu, \cdot) - u_{rb}(\mu, \cdot)\|_{\mathbb{W}} \quad (9)$$

is small. The Galerkin projection method provides a posteriori error bound for coercive problems [6]. The methods used for computing the reduced basis are mainly two: the POD and the Greedy Algorithm [6]. A disadvantage of the Galerkin projection approach is that, for complex problems, like Navier-Stokes equations [16, 26–29], there are no error bounds and the error decreases slowly.

2.2 | Non-intrusive methods

In this section, we briefly introduce some of the most relevant non-intrusive ROM techniques.

2.2.1 | Physics informed neural networks

A fully connected NN with lengths l_0, l_1, \dots, l_L is a parametrized function $h : x \in \mathbb{R}^{l_0} \rightarrow y \in \mathbb{R}^{l_L}$ which has the following form

$$\begin{cases} x_0 = x, \\ x_i = f(A_i x_{i-1} + b_i), & \text{with } i = 1 \dots L-1, \\ x_L = g(A_L x_{L-1} + b_L), \\ y = x_L, \end{cases} \quad (10)$$

where $A_i \in \mathbb{R}^{l_{i-1} \times l_i}$, $b_i \in \mathbb{R}^{l_i}$ are learnable parameters and f is a fixed nonpolynomial function called activation function. The function g is typically the identity for regression problems and the *softmax* function for classification problems [7]. NNs are useful as they provide an approximation u_θ of any continuous function u on compact sets:

$$u_\theta : \mathbb{R}^{\dim(\mathbb{P})+4} \rightarrow \mathbb{R}^n. \quad (11)$$

A PINN [1, 7] is a NN which is trained using the following loss function:

$$\begin{aligned} \sum_{\mu \in \mathbb{P}} \sum_{(x,y,z) \in \Omega(\mu)} \sum_{t \in (0,T]} \|\mathcal{L}(u_\theta(\mu, t, x, y, z), t, x, y, z)\| + \sum_{\mu \in \mathbb{P}} \sum_{(x,y,z) \in \partial\Omega(\mu)} \sum_{t \in (0,T]} \|\mathcal{B}(u_\theta(\mu, t, x, y, z), t, x, y, z)\| + \\ \sum_{\mu \in \mathbb{P}} \sum_{(x,y,z) \in \Omega(\mu)} \|\mathcal{T}(u_\theta(\mu, 0, x, y, z), \mu, x, y, z)\|. \end{aligned} \quad (12)$$

The sum above is computed using Monte Carlo sampling [1, 7]. In the offline phase, the NN is trained using the backpropagation algorithm [1, 7].

2.2.2 | Purely data-driven neural networks

When a full-order database is available, for example, after a FOM has been run, we can approximate u as a function $u_\theta : \mathbb{R}^{\dim(\mathbb{P})} \rightarrow \mathbb{R}^{M \cdot n \cdot N_T}$, where N_T is the number of time instants, n is the dimension of the output, and θ are the parameters of the NN. To train the NN, we adopt a simple L_2 loss

$$\min_{\theta \in \mathbb{R}^{\dim(\theta)}} \|u - u_\theta\|_2^2. \quad (13)$$

In this case when fed with a new parameter in the online phase, the NN will output the entire solution on the discretized domain. This approach is typically faster than running a PINN but it does not support the computation of the solution in points different from the ones of the original discretization.

2.2.3 | Radial basis functions

An RBF is an interpolation method that has the following form

$$g(x) = \sum_{i=1}^n w_i \psi(\|x - x_i\|) + \sum_{j=1}^m c_j p_j(x), \quad (14)$$

where $x_i \in \mathbb{R}^m$ are the interpolation points and w_i are proper weights with $i = 1, \dots, n$ and ψ is a radial function. Moreover, the function $p_j(x)$ outputs the j -th coordinate of x . The weights w_i are learned by solving the following linear system:

$$\begin{cases} \sum_{i=1}^n w_i \psi(\|x_k - x_i\|) + \sum_{j=1}^n z_j p_j(x_k) = y_k & k = 1, \dots, n, \\ \sum_{i=1}^n w_i p_j(x_i) = 0 & j = 1, \dots, m. \end{cases} \quad (15)$$

For small systems, RBF are faster than NNs based techniques, but they cubically scale with the number of interpolation points, and may have high memory demands [8].

2.2.4 | Dynamic mode decomposition

Let us suppose to have a sequence of samples x_1, \dots, x_{N_T} of a time series where $x_k \in \mathbb{R}^n$ for $k = 1, \dots, N_T$. We want to approximate the Koopman operator A :

$$x_{k+1} = Ax_k. \quad (16)$$

To achieve this, we build two matrices

$$S_1 = [x_1, \dots, x_{N_T}], \quad S_2 = [x_2, \dots, x_{N_T-1}]. \quad (17)$$

We would like to find $A \in \mathbb{R}^{n \times n}$ such that

$$\|S_2 - AS_1\| \quad (18)$$

is minimized. However there may be two problems:

- the x_k may contain some noise, so to solve the system as it is may lead to overfitting problems;
- if n is large managing A could be prohibitive.

For this reason it is computed a reduced approximation of A :

$$\tilde{A} = UU^T S_2 V \Sigma^{-1} U^T, \quad (19)$$

where $S_1 = U \Sigma V^T$ is a truncated Singular Value Decomposition. There are extensions of DMD that account for the presence of parameter dependent time series [7, 24].

2.2.5 | Gaussian process regression

A Gaussian process [7] is a non parametric model that depends on a positive definite kernel K . Suppose that, given $x_1, \dots, x_N \in \mathbb{R}^m$ and $y_1, \dots, y_N \in \mathbb{R}$, we want to model a function f such that $\sum_{i=1}^N |f(x_i) - y_i|^2$ is minimized. A Gaussian process assumes that

$$\begin{cases} y(x) = f(x) + \epsilon, & \epsilon \sim \mathcal{N}(0, \sigma), \\ [f(x_i)_{i=1 \dots N}] \sim \mathcal{MNN}(g(x_i)_{i=1 \dots N}, K(x_i, x_j)_{i,j=1 \dots N}), \end{cases} \quad (20)$$

where $g : \mathbb{R}^m \rightarrow \mathbb{R}$ and $\sigma > 0$. It can be proven [30] that for new samples $\bar{y}_i, i = 1 \dots l$ we have that

$$\bar{y} \sim \mathcal{MNN}\left(\hat{\mathbf{K}}[\mathbf{K} + \sigma^2 I]^{-1} \mathbf{y}, \bar{\mathbf{K}} - \hat{\mathbf{K}}[\mathbf{K} + \sigma^2 I]^{-1} \hat{\mathbf{K}}^T\right), \quad (21)$$

where

$$\bar{\mathbf{y}} = \{\bar{y}_i\}_{i=1\dots l}, \quad (22)$$

$$\mathbf{y} = \{y_i\}_{i=1\dots N}, \quad (23)$$

$$\mathbf{K} = k(x_i, x_j)_{i,j=1\dots N}, \quad (24)$$

$$\hat{\mathbf{K}} = k(\bar{x}_i, x_j)_{i=1\dots l, j=1\dots N}, \quad (25)$$

$$\bar{\mathbf{K}} = k(\bar{x}_i, \bar{x}_j)_{i,j=1\dots l}. \quad (26)$$

2.3 | Geometrical parametrization and active subspace

When Ω depends on μ the problem becomes more complex because if we want to use the same discretization for every different parameter we need to ensure that it does not produce holes or overlapping elements for every domain. As such, there exists parametrization techniques like Free Form Deformation (FFD) [7, 21] or RBF [7, 10] that are able to parametrize a reference geometry and to morph it without errors.

In this context, before proceeding with the development of the surrogate model, it is important to try to reduce the parameter space dimension. At this aim, a technique that can be used is the AS method [7, 18, 31]. Let us suppose to have N samples, x_1, \dots, x_N , where the generic sample $x_i \in \mathbb{R}^m$ and a continuous function $f : \mathbb{R}^m \rightarrow \mathbb{R}^n$ which is also differentiable with continuous derivatives. Then we can construct the matrix C

$$C = \frac{1}{N} \sum_{i=1}^N \nabla f(x_i)^T \nabla f(x_i). \quad (27)$$

C is symmetric positive definite so it is diagonalizable and admits a decomposition $C = V\Sigma V^T$. Assuming that the eigenvalues of Σ are sorted in decreasing order one could perform dimensionality reduction by considering $z_i = Wx_i$ where W are the first k columns of V .

3 | ACADEMIC BENCHMARKS

Now we are going to test the performance of the ROM approaches described in the previous section in terms of efficiency and accuracy against some relevant academic test cases. All the simulations were run on an Intel Core i5-1145G7 2.60 GHz processor.

Hereinafter we omit to explicitly report the dependency of the solution on μ at the aim to make the notation lighter.

As first test case we consider a classic fluid dynamics problem, the lid driven cavity [16]. Here we consider some of the data-driven ROMs presented in Section 2.2. The FOM is given by the Navier–Stokes equations:

$$\begin{cases} \mathbf{u}_t + (\mathbf{u} \cdot \nabla)\mathbf{u} - \nu \Delta \mathbf{u} + \nabla p = 0 & \text{in } \Omega \times (0, T), \\ \nabla \cdot \mathbf{u} = 0 & \text{in } \Omega \times (0, T), \end{cases} \quad (28)$$

where \mathbf{u} is the velocity vector field, p is the pressure, and ν is the kinematic viscosity coefficient.

The computational domain Ω is $[0, 1]^2$. We consider the bottom left corner of the square as the origin of the axes. We impose a no slip boundary condition on the lower and lateral walls. At the upper wall we prescribe a Dirichlet boundary condition $\mathbf{u}(t, x, 1) = (1, 0)$ (see Figure 1). We consider a structured mesh consisting of 105×105 square cells. We start the simulations from fluid at rest and we set $\Delta t = 0.005$.

We set $\mu = \nu$, that is, the parameter of our problem coincides with the kinematic viscosity. We have $\mathbb{P} = [0.001, 0.01]$. We consider a uniform sampling of \mathbb{P} consisting of 600 samples to train our ROM. For every training parameter, a FOM simulation was computed by collecting 100 time-dependent snapshots over the time interval of ten nondimensional time units ($T = 10$).

All the FOM simulations are performed using OpenFOAM (<https://www.openfoam.com/>), an open source finite volume C++ library widely used by commercial and academic organizations. On the other hand, the ROM computations are

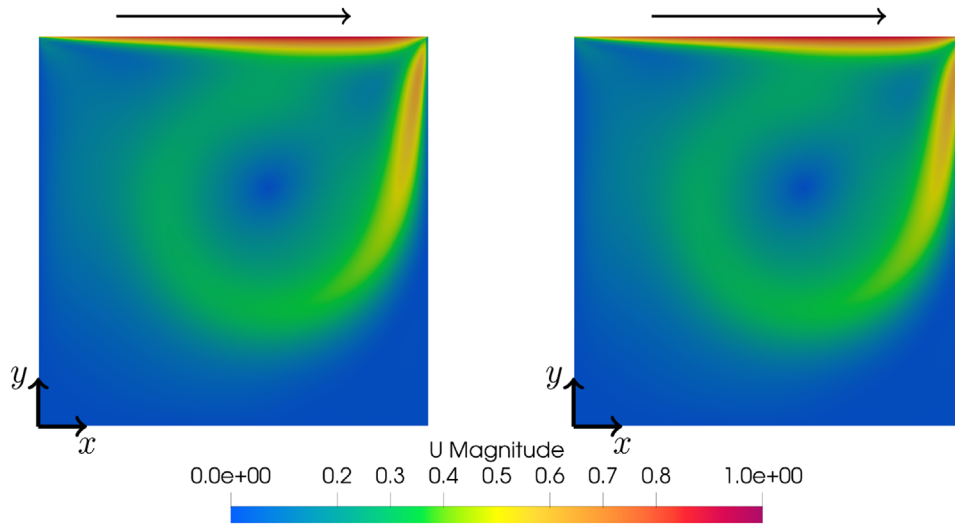


FIGURE 1 Lid driven cavity: velocity magnitude at the final time of the simulation, that is, $T = 10$, for $\mu = 0.001$ computed by FOM (left) and by DMD + RBF (right). The arrow denotes the moving wall. DMD, dynamic mode decomposition; FOM, full order model; RBF, radial basis functions.

TABLE 1 Lid driven cavity: performance of ROM models related to the reconstruction of the streamwise component of \mathbf{u} .

	NN	DMD + NN	DMD + GPR	DMD + RBF
Train relative error	7×10^{-2}	2.4×10^{-5}	6.9×10^{-5}	7.2×10^{-6}
Test relative error	6.6×10^{-2}	1.8×10^{-5}	6.3×10^{-5}	7.0×10^{-6}
Speedup	7×10^4	3×10^2	1×10^2	3×10^2

Note: The test value is $\mu = 0.001$.

Abbreviations: DMD, dynamic mode decomposition; GPR, Gaussian process regression; NN, neural networks; RBF, radial basis functions

carried out using some python-based packages: PyDMD (<https://pydmd.github.io/PyDMD/>), PyTorch (<https://pytorch.org/>) and scikit-learn (<https://scikit-learn.org/stable/>).

We take $\mu = 0.001$ (in the range under consideration but not in the training set) to evaluate the performance of the parametrized ROM. The results related to the reconstruction of streamwise component of \mathbf{u} are summarized in Table 1. We consider 400 modes for the DMD.

We see that the DMD + RBF model is the most accurate whilst the NN one is the most efficient. In Figure 1 we show a qualitative comparison between FOM and DMD + RBF.

As a second test case we consider the two dimensional flow past a cylinder [32]. The FOM is the same of the previous benchmark; it is given by Equations (3). Here we test both intrusive and non-intrusive ROM approaches. The computational domain is a 2.2×0.41 rectangular channel with a cylinder of radius 0.05 centered at (0.2, 0.2), when taking the bottom left corner of the channel as the origin of the axes. See Figure 2. We impose a no slip boundary condition on the upper and lower wall and on the cylinder. At the inflow and the outflow we prescribe the following velocity profile

$$\mathbf{u}(t, 0, y) = \left(\frac{6}{0.41^2} y(0.41 - y), 0 \right), \quad y \in [0, 0.41]. \quad (29)$$

We consider a grid consisting of 6000 triangular cells. We start the simulations from fluid at rest and we set $\Delta t = 0.01$.

We set $\mu = \nu$ and $\mathbb{P} = [0.01, 0.1]$. We consider a uniform sampling of \mathbb{P} consisting of 600 samples to train our ROM. For every training parameter, a FOM simulation was computed by collecting 100 time-dependent snapshots over the time interval of eight nondimensional time units ($T = 8$).

All the FOM simulations are performed by using FEniCS (<https://fenicsproject.org/>), an open source finite element python library widely used by academic organizations. On the other hand, the ROM computations are carried out using the python packages Rbnics (<https://www.rbnicsproject.org/>), PyTorch (<https://pytorch.org/>) and scikit-learn (<https://scikit-learn.org/stable/>).

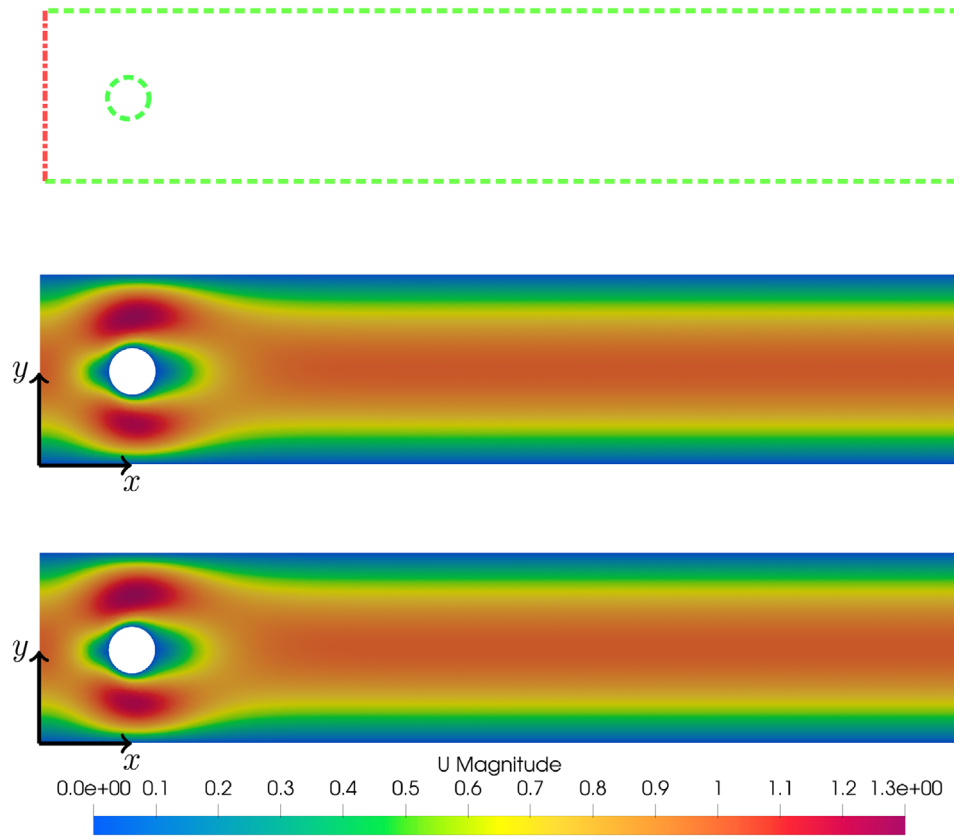


FIGURE 2 Flow past a cylinder: geometry (first row, the inflow boundary is colored in red, the wall boundary in green and the outflow boundary in violet), velocity magnitude at $t = 1$ s for $\mu = 0.5$ computed by FOM (second row) and by the POD-Galerkin model (third row with 15 basis functions). FOM, full order model; POD, proper orthogonal decomposition.

TABLE 2 Flow past a cylinder: performance of ROM models related to the reconstruction of magnitude of \mathbf{u} .

	POD-GALERKIN ($N_{rb} = 15$)	POD-GALERKIN ($N_{rb} = 5$)	GPR	RBF
Train relative error	1.6×10^{-6}	9.1×10^{-3}	1.8×10^{-4}	8.8×10^{-17}
Test relative error	5.36×10^{-5}	3.26×10^{-2}	5.05×10^{-3}	6.82×10^{-3}
Speedup	1.5	2	3×10^4	1×10^5

Note: The test value is $\mu = 0.5$.

Abbreviations: GPR, Gaussian process regression; RBF, radial basis functions; POD, proper orthogonal decomposition.

We take $\mu = 0.5$ (in the range under consideration but not in the training set) to evaluate the performance of the parametrized ROM. The results related to the reconstruction of the magnitude of \mathbf{u} are summarized in Table 2. In this case, we see that the Galerkin projection (with 15 basis functions) is the most accurate whilst the RBF approach is the most efficient. Moreover it appears evident that, in general, non intrusive ROMs are significantly faster than the Galerkin projection approach although they have in average a lower accuracy. In Figure 2 we show a qualitative comparison between FOM and Galerkin ROM model for $\mu = 0.5$.

As a third and last academic test case we consider an example of geometrical parametrization for a three-dimensional domain. In particular, we refer to the parametrized Stanford Bunny [33] problem. We consider the following Poisson equation

$$\Delta u = e^{-\|x-x_n\|^2} \quad \text{in } \Omega(\mu), \quad (30)$$

where x_n is the vector identifying the barycenter of the domain. We impose the following boundary condition

$$u(x) = e^{-\|x-x_n\|} \quad \text{on } \partial\Omega(\mu). \quad (31)$$

TABLE 3 Stanford Bunny test case: performance of ROM models related to the reconstruction of the solution of the Poisson equation on the parametrized domain.

	RBF	GPR	NN
Train relative error	4.1×10^{-9}	7.1×10^{-11}	4.3×10^{-2}
Test relative error	4.7×10^{-2}	5.5×10^{-2}	4.7×10^{-2}
Speedup	8×10^4	1.6×10^4	1.6×10^4

Abbreviations: GPR, Gaussian process regression; NN, neural networks; RBF, radial basis functions; ROM, reduced order models.

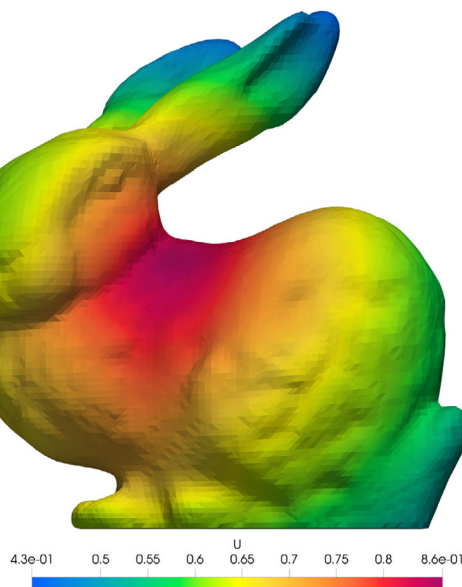


FIGURE 3 Parametrized Stanford Bunny: plot of the solution for a new value of μ .

The mesh consists of 182 184 triangular cells which is parametrized using the FFD method [7]. The parameter μ of our problem represents the deformation of the control points of the FFD map and we set $\mathbb{P} = [-0.4, 0.4]$. We consider a normal distribution of 600 samples to train our ROM.

For every training parameter, the solution of the Poisson equation was computed by using FeniCSx (<https://fenicsproject.org/>).

The ROM results are summarized in Table 3. In this case, we see that the RBF model exhibits the best performance in terms of efficiency and accuracy. In Figure 3 we show the solution obtained for a value of μ in the range under consideration but not in the training set.

4 | TOWARDS REAL WORLD APPLICATIONS: AN INDUSTRIAL CASE

Now we present an example coming from industry in the framework of a research collaboration with Electrolux concerning domestic refrigerator. Such a problem involves a multiphysics scenario and it is useful to show the performance of data-driven ROM approaches for complex cases. This case study is still work in progress, so we limit to provide some preliminary outcomes.

At the full order level, air circulation and heat transfer in fluid and between fluid and surrounding solids in a fridge were numerically studied using the Conjugated Heat Transfer (CHT) method to explore both the natural and forced convection-based fridge model followed by a parametric study based on the ambient temperature T_{amb} , fridge fan velocity v_f , and evaporator temperature T_{ev} . The FOM simulations are performed using OpenFOAM (<https://www.openfoam.com>).

A non-intrusive ROM based on a POD-NN approach is considered to obtain the temperature field at specific parametric locations where the training dataset is purely based on numerical computation. The ROM computations were performed in EZyRB (<https://github.com/mathLab/EZyRB>).

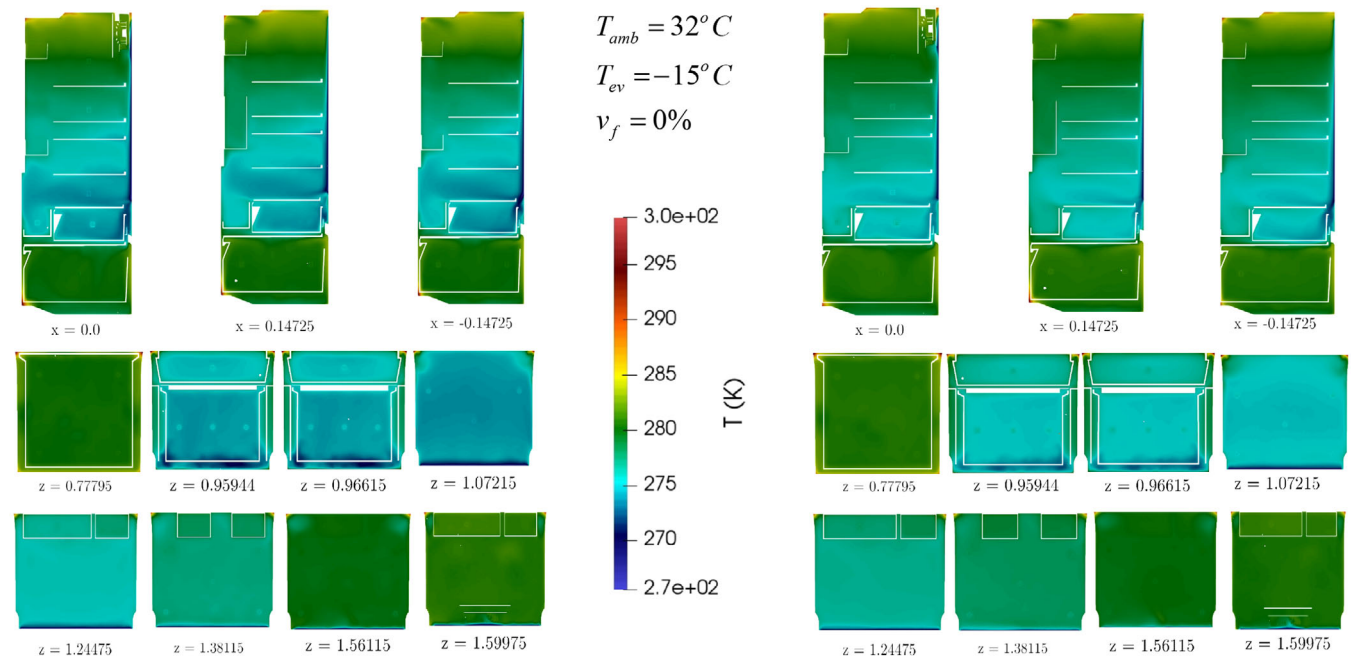


FIGURE 4 Industrial application: temperature in different sections of the fridge system computed by FOM (left) and by POD-ANN (right). FOM, full order model, POD, proper orthogonal decomposition.

To carry out the parametric study, ν_f ranges from 0% to 100% of the full throttle which is 1600 rpm in step of 10% that is, a total of 11 samples, two values of T_{amb} that is, 16°C and 32°C, and four values of T_{ev} that is, -15°C, -7.9°C, -3.25°C and 4°C are considered.

In Figure 4 we show the FOM and ROM solutions obtained for $T_{amb} = 32^\circ\text{C}$, $T_{amb} = -15^\circ\text{C}$ and $\nu_f = 0\%$ in different sections of the fridge system. The comparison is quite satisfactory. This preliminary result gives an indication that data-driven surrogate models can be useful for real-world applications in addition to canonical academic examples.

5 | CONCLUSION

In this paper we have introduced some ROM techniques, both intrusive and non-intrusive. We have tested their performance in terms of efficiency and accuracy against three academic test cases, the lid driven cavity, the flow past a cylinder and the geometrically parametrized Stanford Bunny. Moreover, we also briefly presented some preliminary results related to a more complex case involving an industrial application.

From the brief analysis carried out on academic benchmarks, it appears evident that, within a physically parameterized setting, non-intrusive ROMs are significantly faster although the projection-based ones ensure a greater accuracy especially when a significant number of modes is retained. On the other hand, when a geometrical parametrization is introduced, all the data-driven ROMs tested, RBF, GPR and NN, work very well, with the RBF approach providing slightly better results.

A last important outcome to be underlined is that data-driven surrogate models seem to be very promising also for real-world problems as shown by the industrial example presented.

ACKNOWLEDGMENTS

We acknowledge the support provided by PRIN “FaReX - Full and Reduced order modelling of coupled systems: focus on non-matching methods and automatic learning” project, INdAM-GNCS 2019–2020 projects, PNRR NGE iN- EST “Inter-connected Nord-Est Innovation Ecosystem” project, and PON “Research and Innovation on Green related issues” FSE REACT-EU 2021 project.

ORCID

Gianlugi Rozza  <https://orcid.org/0000-0002-0810-8812>

REFERENCES

1. Demo, N., Strazullo, M., & Rozza, G. (2023). An extended physics informed neural network for preliminary analysis of parametric optimal control problems. *Computers & Mathematics with Applications*, *143*, 383–396.
2. Es-haghi, M. S., Anitescu, C., & Rabczuk, T. (2024). Methods for enabling real-time analysis in digital twins: A literature review. *Computers & Structures*, *297*, 35.
3. Tezzele, M., Demo, N., Stabile, G., Mola, A., & Rozza, G. (2020). Enhancing CFD predictions in shape design problems by model and parameter space reduction. *Advanced Modeling and Simulation in Engineering Sciences*, *7*, 19.
4. Tezzele, M., Demo, N., & Rozza, G. (2016). Efficient reduction in shape parameter space dimension for ship propeller blade design. Proceedings of the VIII international conference on computational methods in marine engineering: Marine 2019, Gothenburg, Sweden, pp. 122–133.
5. Salmoiraghi, F., Ballarin, F., Corsi, G., Mola, A., Tezzele, M., & Rozza, G. (2016). Proceedings of the 7th European Conference on Computational Methods in Applied Sciences and Engineering, Crete Island, Greece, pp. 1013–1031.
6. Hesthaven, J., Rozza, G., & Stamm, B. (2015). *Certified reduced basis methods for parametrized partial differential equations*, SpringerBriefs in Mathematics. Springer.
7. Rozza, G., Stabile, G., & Ballarin, F. (2022). *Advanced reduced order methods and applications in computational fluid dynamics*. SIAM.
8. Buhmann, D. H. (2003). *Radial basis functions: Theory and implementations*. Cambridge University Press.
9. Padula, G., Romor, F., Stabile, G., & Rozza, G. (2024). 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.
10. Ivagnes, A., Demo, N., & Rozza, G. (2024). A shape optimization pipeline for marine propellers by means of reduced order modeling techniques. *International Journal for Numerical Methods in Engineering*, *125*, e7426.
11. Coscia, D., Demo, N., & Rozza, G. (2024). Generative adversarial reduced order modelling. *Scientific Reports*, *14*, 3826.
12. Hajisharifi, A., Girfoglio, M., Quaini, A., & Rozza, G. (2024). A comparison of data-driven reduced order models for the simulation of mesoscale atmospheric flow. *Finite Elements in Analysis and Design*, *228*, 104050.
13. Balzotti, C., Siena, P., Girfoglio, M., Stabile, G., Dueñas-Pamplona, J., Sierra-Pallares, J., Amat-Santos, I., & Rozza, G. (2024). A reduced order model formulation for left atrium flow: An atrial fibrillation case. *Biomechanics and Modeling in Mechanobiology*, *23*, 1411–1429.
14. Ivagnes, A., Tonicello, N., Cinnella, P., & Rozza, G. (2024). Enhancing non-intrusive Reduced Order Models with space-dependent aggregation methods. *Acta Mechanica*.
15. Siena, P., Africa, P. C., Girfoglio, M., & Rozza, G. (2024). On the accuracy and efficiency of reduced order models: Towards real-world applications. *Advances in Applied Mechanics*, 245–288.
16. Prusak, I., Torlo, D., Nonnino, M., & Rozza, G. (2024). An optimisation-based domain-decomposition reduced order model for parameter-dependent non-stationary fluid dynamics problems. *Computers & Mathematics with Applications*, *166*, 253–268.
17. Salavatidezfouli, S., Nikishova, A., Torlo, D., Teruzzi, M., & Rozza, G. (2024). *Quantitative sustainability*. Springer.
18. Meneghetti, L., Demo, N., & Rozza, G. (2023). A dimensionality reduction approach for convolutional neural networks. *Applied Intelligence*, *53*, 22818–22833.
19. Hinze, M., Kutz, J. N., Mula, O., & Urban, K. (2023). Model order reduction and applications: Cetraro, Italy. In Falcone, M., & Rozza, G. (Eds.), *Lecture Notes in Mathematics*. Springer.
20. Gonnella, I. C., Hess, M. W., Stabile, G., & Rozza, G. (2023). A two-stage deep learning architecture for model reduction of parametric time-dependent problems. *Computers & Mathematics with Applications*, *149*, 115–127.
21. Siena, P., Girfoglio, M., Ballarin, F., & Rozza, G. (2023). A Data-driven reduced order modelling for patient-specific hemodynamics of coronary artery bypass grafts with physical and geometrical parameters. *Journal of Scientific Computing*, *94*, 38.
22. Regazzoni, F., Pagani, S., Salvador, M., & Quarteroni, A. (2024). Learning the intrinsic dynamics of spatio-temporal processes through Latent Dynamics Networks. *Nature Communications*, *15*, 1834.
23. Papapicco, D., Demo, N., Girfoglio, M., Stabile, G., & Rozza, G. (2022). The Neural Network shifted-proper orthogonal decomposition: A machine learning approach for non-linear reduction of hyperbolic equations. *Computer Methods in Applied Mechanics and Engineering*, *393*, 114687.
24. Andreuzzi, F., Demo, N., & Rozza, G. (2023). A dynamic mode decomposition extension for the forecasting of parametric dynamical systems. *SIAM Journal on Applied Dynamical Systems*, *22*, 2432–2458.
25. Quarteroni, A., & Valli, A. (1994). *Numerical approximation of partial differential equations*, Springer Series in Computational Mathematics. Springer.
26. Girfoglio, M., Quaini, A., & Rozza, G. (2023). A hybrid projection/data-driven reduced order model for the Navier-Stokes equations with nonlinear filtering stabilization. *Journal of Computational Physics*, *486*, 112127.
27. Sheidani, A., Salavatidezfouli, S., Stabile, G., Gerdroodbary, M. B., & Rozza, G. (2023). Assessment of icing effects on the wake shed behind a vertical axis wind turbine. *Physics of Fluids*, *35*, 095135.
28. Mola, A., Giuliani, N., Crego, O., & Rozza, G. (2023). A unified steady and unsteady formulation for hydrodynamic potential flow simulations with fully nonlinear free surface boundary conditions. *Applied Mathematical Modelling*, *122*, 322–349.

29. Khamlich, M., Pichi, F., & Rozza, G. (2021). Model order reduction for bifurcating phenomena in fluid-structure interaction problems. *International Journal for Numerical Methods in Fluids*, 94, 1611–1640.
30. Rasmussen, C. E., & Williams, C. K. I. (2006). *Gaussian processes for machine learning*. The MIT Press.
31. Romor, F., Tezzele, M., & Rozza, G. (2024). A local approach to parameter space reduction for regression and classification tasks. *Journal of Scientific Computing*, 99, 83.
32. Rozza, G., Ballarin, F., Scandurra, L., & Pichi, F. (2024). *Real time reduced order computational mechanics, SISSA Springer Series*. Springer.
33. Curless, B., & Levoy, M. (1996). A volumetric method for building complex models from range images. In Proceedings of the 23rd annual conference on Computer graphics and interactive techniques, New Orleans, United States (pp. 303–312). Association for Computing Machinery.

How to cite this article: Padula, G., Girfoglio, M., & Rozza, G. (2024). A brief review of reduced order models using intrusive and non-intrusive techniques. *Proceedings in Applied Mathematics and Mechanics*, 24, e202400210. <https://doi.org/10.1002/pamm.202400210>