

# Chapter 1

## Reduced-order modelling

Models are useful to understand a given problem or physical phenomena and together with the computers we could predict, optimize, control, etc. some aspects we are interested in. But sometimes the model can contain a lot of degrees of freedom (independent variables or parameters) due to his nature, so trying to solve it can take a lot computing time (it can last years or even more) and/or heavy memory storage. The physical data (numerical simulations, measurements experiments) can be strongly correlated so there are redundancies in space, time or in the parameters. Then we can take advantages of these redundancies in order to reduce computing times or storage, and that can be achieved using *Reduced-Order Models*. Reduced-order models aim at:

- Taking advantages of redundancies
- Identifying “genuine” degrees of freedom
- Giving low dimensional approximations
- Preserving a satisfactory accuracy
- Decreasing the computational resources (time and storage)

The reduced-order models try to use only the degrees of freedom which contain the most of the information.

## 1. REDUCED-ORDER MODELLING

---

### 1.1 Introduction

The most convenient part of the reduced-order models is that they can be used to describe a physical phenomena using less quantity of degrees of freedom than the original required. These can be done because it extracts the most relevant dof's and then use them to express a new solution when the parameters change. This allow to solve really quick new problems once the ROM has been created.

In order to get the most relevant dof's can be used several techniques. These are some of the techniques the ROMs are based on:

- Physical tuition, phenomenological modelling, empirical constructs
- Interpolation (polynomial or Kriging)
- Either *singular values decomposition* (SVD) or *high-order singular value decomposition* (HOSVD)
- *Proper orthogonal decomposition* (POD) possibly combined with some projection of the involved equations

In this case we are going to use the POD combined with a Galerkin projection. The general idea to create a POD based ROM is given a training set (in our case will be a bunch of solutions) get a orthogonal basis (using POD) and then use the basis (here we choose how many dof's for the order-reduced model) to accurately represent the solution for a new problem. This sounds good but like any method it has its drawbacks, some of them:

1. Creating the ROM could require a lot of computing time
2. The goal is to create a cheap ROM (with less dof's) and accurate as possible (a good trade-off has to be found between these two variables)
3. When using a ROM with a non-linear problem the required computational cost is not decreased as to justify the use of the ROM
4. The ROM's lack robustness so it can only be used with new problems with a small change in the parameters, when the change is big the ROM has to be constructed again

## 1.2 Reducing a variational problem

Suppose that the unknown variable  $\mathbf{U}_{n+1}$  for a variational, that has been discretized in space and time, problem can be written as follows

$$\mathbf{F}(\mathbf{U}_{n+1}, \mathbf{U}_n, \mathbf{U}_{n-1}, \dots) := \mathbf{A}(\mathbf{U}_{n+1})\mathbf{U}_{n+1} - \mathbf{B}_n(\mathbf{U}_n)\mathbf{U}_n - \mathbf{B}_{n-1}(\mathbf{U}_{n-1})\mathbf{U}_{n-1} - \dots - \mathbf{C} = 0 \quad (1.1)$$

If  $M$  is the number of unknowns then  $\mathbf{U}_{n+1}, \mathbf{U}_n, \mathbf{U}_{n-1}, \dots, \mathbf{F}, \mathbf{C} \in R^M$ . The matrices  $\mathbf{A}, \mathbf{B}_n, \mathbf{B}_{n-1}, \dots, \mathbf{B}_1 \in R^{M \times M}$  and as shown in eq. 1.1 these matrices may be non-linear because they depend on the values of  $\mathbf{U}$ . We can group the constant terms and the ones related to previous time steps like

$$\mathbf{R}(\mathbf{U}_n, \mathbf{U}_{n-1}, \dots) := \mathbf{B}_n(\mathbf{U}_n)\mathbf{U}_n + \mathbf{B}_{n-1}(\mathbf{U}_{n-1})\mathbf{U}_{n-1} + \dots + \mathbf{C} \quad (1.2)$$

As we said this is a non-linear matrix because it depends on  $\mathbf{U}$  and to solve the problem we need to linearize it. To do so we can use Picard's or Newton's method. For this work Picard's method was used because its radius of convergence is larger than the Newton's method for the incompressible Navier-Stokes equations. Using Picard's method the problem can be solved iteratively as follows

$$\mathbf{A}(\mathbf{U}_{n+1}^{i-1})\mathbf{U}_{n+1}^i = \mathbf{R}(\mathbf{U}_n, \mathbf{U}_{n-1}, \dots) \quad (1.3)$$

To construct the reduced-order model the system of equations in eq. 1.3 can be projected onto a low-dimensional subspace  $\mathcal{U} \subset R^M$  and  $\mathbf{U}$  can be approximated by

$$\mathbf{U} \simeq \tilde{\mathbf{U}} = \boldsymbol{\Phi}_u \mathbf{U}_\Phi \quad (1.4)$$

Here  $\boldsymbol{\Phi}_u$  is the basis for  $\mathcal{U}$  and  $\boldsymbol{\Phi}_u \in R^{M \times N}$ .  $N$  is the dimension of the reducer-order model so  $N \leq M$ .  $\mathbf{U}_\Phi$  are the components for the solution in the reference system defined by  $\boldsymbol{\Phi}_u$  and  $\mathbf{U}_\Phi \in R^N$ . The basis  $\boldsymbol{\Phi}_u$  can be obtained using Proper Orthogonal Decomposition (POD), as we see later.

A very important aspect is that the reduced order basis functions fulfill the boundary conditions of the full order model, so any linear combination of the functions also

## 1. REDUCED-ORDER MODELLING

---

fulfill this requirement. Then we can use our ROM on a new problem and be sure that the boundary conditions are correct.

Substituting eq. 1.4 in 1.3 we get a system with  $M$  equations and  $N$  unknowns:

$$\mathbf{A}(\mathbf{U}_{n+1}^{i-1})\Phi_u \mathbf{U}_{\Phi,n+1}^i = \mathbf{R}(\mathbf{U}_n, \mathbf{U}_{n-1}, \dots) \quad (1.5)$$

we can try solve this overdetermined system with least squares which leads to

$$\Phi_u^T \mathbf{A}(\mathbf{U}_{n+1}^{i-1})\Phi_u \mathbf{U}_{\Phi,n+1}^i = \Phi_u^T \mathbf{R}(\mathbf{U}_n, \mathbf{U}_{n-1}, \dots) \quad (1.6)$$

where  $\mathbf{U}_{\Phi,n+1}^i$  minimize the error given by the matrix  $\mathbf{A}$  at time  $n$  and iteration  $i - 1$

$$\mathbf{U}_{\Phi,n+1}^i = \arg \min_{\mathbf{U}_{\Phi} \in \mathcal{U}} \|\Phi_u \mathbf{U}_{\Phi} - \mathbf{A}(\mathbf{U}_{n+1}^{i-1})^{-1} \mathbf{R}(\mathbf{U}_n, \mathbf{U}_{n-1}, \dots)\|_{\mathbf{A}(\mathbf{U}_{n+1}^{i-1})} \quad (1.7)$$

The previous process can only be applied when the matrix  $\mathbf{A}$  is symmetric and positive definite otherwise a Petrov-Galerkin is required to ensure the optimality of the recovered solution 1.3.

### 1.3 Stabilized Navier-Stokes equations

Describe the stabilized finite element approximation of the Navier-Stokes equations. This can be moved to another chapter.

### 1.4 Proper orthogonal decomposition

Describe the POD

## 1.5 Explicit reduced-order modelling model for the Navier-Stokes equations

### 1.5.1 POD based ROMs applied in non-linear problems

The drawback when solving non-linear problems (for example the incompressible Navier-Stokes equations)

Way to approach the last problem (hyper-reduction)

### 1.5.2 Explicit reduced-order modelling

Describe the explicit reduced-order model