Quasi real time simulation of a capsule in flow using Reduced Order Models

Toufik Boubehziz

To cite this version:

Toufik Boubehziz. Quasi real time simulation of a capsule in flow using Reduced Order Models. Biomechanics [physics.med-ph]. Université de Technologie de Compiègne, 2022. English. NNT : 2022COMP2678 . tel-03771823

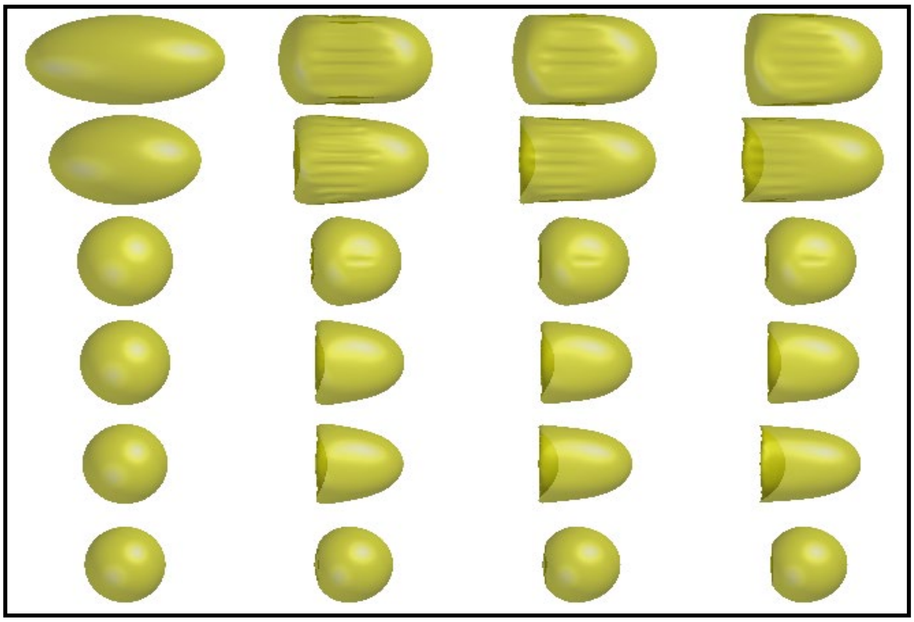
HAL Id: tel-03771823
https://theses.hal.science/tel-03771823

Submitted on 7 Sep 2022

Par **Toufik BOUBEHZIZ**

Simulation en quasi temps réel d'une capsule sous écoulement grâce a des Modèles d'Ordre Réduit

Thèse présentée
pour l'obtention du grade
de Docteur de l'UTC



Soutenue le 14 avril 2022
Spécialité : Biomécanique - Équations aux dérivées partielles et méthodes numériques : Unité de Recherche en Biomécanique et Bioingénierie (UMR-7338)
D2678

THÈSE
Présentée par

Toufik Boubehziz

Pour l'obtention du grade de

Docteur de l'Université de Technologie de Compiègne

Spécialités : 1. Biomécanique 2. Équations aux dérivées partielles et méthodes numériques

---

Simulation en quasi temps réel d'une capsule sous écoulement grâce à des Modèles d'Ordre Réduit

---

Soutenue le 14 avril 2022 devant le jury composé de :

M. Alain Rassineux	**Université de Technologie de Compiègne**	**Président du jury**
M. Rodolphe Le Riche	**Mines de Saint Etienne**	**Rapporteur**
M. Jose Maria Fullana	**Sorbonne Université**	**Rapporteur**
Mme Anne–Virginie Salsac	**Université de Technologie de Compiègne**	**Directrice de Thèse**
M. Pierre Villon	**Université de Technologie de Compiègne**	**Invité**

"Every accomplishment starts with the decision to try."
— John F. KENNEDY

In the memory of Azzedine Boubehziz
In the memory of Boudjemaa Belhani

Acknowledgements

Beyond the scientific aspect, this work wasn't possible without the support of wonderful persons whom I cannot thank enough. I will therefore attempt to express my gratitude to all the people who, in one way or another, have helped me to carry out the work presented in this manuscript.

First of all, I would like to express my gratitude to my supervisor Anne-Virginie Salsac for the opportunity that gave me and for her valuable advice. My thanks go to the reading committee, Rodolphe Le Riche and Jose Maria Fullana, who reported my thesis and for their useful remarks and comments. Many thanks also go to Alain Rassineux and Pierre Villon for accepting to be part of my thesis jury and for their interesting and relevant comments on my PhD work.

Special thanks to Martin Quinson, Piotr Breitkopf, Imad Rida, Xu Qu Hu, and Badr Kaoui for their availability, advice, and support. I would like to thank also my fellow labmates Adlan Merlo, Woorak Choi, Clément Bielinksi, Nicolas Grandmaison, Rachid Djellali, and Anne Le-Goff at Bio-engineering and Biomedical (BMBI) laboratory for the endless scientific and no scientific discussions in the last three years.

My final appreciation goes to my family members for their full support throughout all these years.

Abstract

The motion of a liquid-filled microcapsule flowing in a microchannel is a complex problem to simulate. Two innovative reduced-order data-driven models are proposed to replace the Fluid-Structure Interaction (FSI) model using a collected database from high-fidelity simulations. The objective is to replace the existing Full Order Model (FOM) with a fast-simulation model that can simulate the capsule deformation in flow at a low cost in terms of time and calculation. The first model consists in building from a space-time-parameter datacube a reduced model to simulate the deformation of the microcapsule for any admissible configuration of parameters. Time evolution of the capsule deformation is treated by identifying the nonlinear low-order manifold of the reduced variables. Then, manifold learning is applied using the Diffuse Approximation (DA) method to predict capsule deformation for a query configuration of parameters and a chosen time discretization. The second model is based on rewriting the FSI model under the form of a reduced-order dynamic system. In this latter, the spectral displacement and velocity coefficients are related through a dynamic operator to be identified. To determine this operator, we suggest the use of a dynamic mode decomposition approach.

Numerical validations prove the reliability and stability of the two new models compared to the high order model. A software application has been developed to explore the capsule deformation evolution for any couple of admissible parameters.

Keywords: data-driven model, model order reduction, proper orthogonal decomposition, manifold learning, diffuse approximation, microcapsule suspension, dynamic mode decomposition.

Résumé

La déformation d'une capsule en écoulement dans un canal micro-fluidique est un problème compliqué à simuler numériquement. Nous proposons deux modèles innovants de pilotage de données d'ordre réduit pour simuler le problème spatio-temporel à partir d'une base de données collectée des simulations réalisées avec un modèle d'ordre élevé. L'objectif est de remplacer le modèle numérique haute-fidèlité existant par un modèle d'ordre réduit capable de simuler l'évolution de déformation des capsules en écoulement à faible coût en temps et en calcul. Le premier modèle consiste à construire à partir d'un cube de données espace-temps-paramètre un modèle réduit pour simuler la déformation de la microcapsule pour n'importe quelle configuration admissible de paramètres. La prédiction de l'évolution temporelle de la capsule pour une configuration donnée de paramètres et un pas de discrétisation temporelle choisi se fait à l'aide d'un apprentissage sur des variétés du modèle réduit. Le deuxième modèle se base sur l'idée de réécrire le problème sous forme d'un système dynamique d'ordre réduit dans lequel les coefficients spectraux des déplacements et les champs des vitesses sont reliés à travers d'un opérateur dynamique à identifier. Pour déterminer ce dernier, nous suggérons l'utilisation d'une approche de décomposition en modes dynamiques.

Des validations numériques confirment la fiabilité et stabilité des deux nouveaux modèles par rapport au modèle d'ordre élevé. Une application informatique est également mise au point afin d'explorer l'évolution de déformation des capsules pour toute configuration de paramètres admissibles.

mots-clés: Pilotage des données, modèle d'order reduit, decomposition orthogonale propre, apprentissage geometrique, approximation diffuse, microcapsule en suspension, décomposition en modes dynamiques.

Contents

List of Figures

Chapter 1

General introduction

In the last decades, an unprecedented increase in computation ability has been achieved until the point of simulating complex systems in a high-fidelity manner. Such advancement is driving a paradigm shift across many fields such as health, engineering, and technology more rapidly than ever before. However, the high-fidelity simulation aspect of these systems becomes computationally expensive and, thus, makes intractable many decision-making applications, such as code parallelization with High-Performance Computing (HPC) (Nielsen, 2016), inverse problems (Belov, 2012), and uncertainty quantification with Monte-Carlo-like approaches (Hastie *et al.*, 2009). For most applications, even with HPC facilities, the resolution of high dimensional multi-coupled problems may be obtained in an unreasonable amount of time and computation.

To compensate the computational expense issue, many surrogate models have been developed, such as Model Reduction order (MOR) and machine learning that extract meaningful spatiotemporal patterns of a stored database from the Full Order Model (FOM) and use it in future case studies. This leads to less redundant computations while the model continues to seek the best result with the lowest cost possible. The resolution of physical systems using this kind of approaches has gained significant interest in the last couple of years, especially with the emergence of Physics-Informed Neural Networks (PINN), where the ANN is trained using a loss function that includes physical information (Raissi *et al.*, 2019). The data-driven model order reduction techniques are a popular class of 'machine learning' methods that use data generated from a time evolution FOM (Benjamin & Karen, 2015, 2016; Pawar *et al.*, 2019; Xiao *et al.*, 2017). Reduced Order Models (ROMs) usually perform a dimensionality reduction through suitable reduced bases. This can be achieved through different approaches such as the Proper Orthogonal Decomposition (POD) (Cordier, 2008; Benner *et al.*, 2015; Silva & Alvaro, 2015), piecewise tangential interpolation (Gallivan *et al.*, 2004), Proper Generalized Decompositions (PGD) (Chinesta *et al.*, 2011, 2010; Ghnatios *et al.*, 2012), Empirical Interpolation Methods (EIM) (Maxime *et al.*, 2004; Chaturantabut & Sorensen, 2010; Xiao *et al.*, 2014) or via different greedy procedures (Lappano *et al.*, 2016).

This work aims to treat the dynamics of dilute suspensions of a liquid core microcapsule in a microfluidic channel. Encapsulation technology has promising applications in the bioengineering and pharmaceutical industry sectors. It is used to protect and deliver active substances from the external suspending fluid until their liberation in a targeted location (del Burgo *et al.*, 2015; Rabanel *et al.*, 2009). Understanding, modeling, and tracking the capsule behavior in such a context remains a challenging problem in current research topics, whether by running experiments or by simulating the problem numerically.

From a numerical point of view, a dilute suspension of spherical micrometric capsules in a microfluidic channel is a complex three-dimensional inertialess fluid-structure interaction problem that interestingly depends on two main variables: the capillary number Ca and the confinement ratio a/ℓ. Ca is a non-dimensional number that estimates the order of magnitude of the viscous forces acting on the capsule with respect to the elastic forces that build up in the membrane, and a/ℓ is the ratio between the initial capsule diameter and the channel width.

This thesis is a part of a European project, "MultiphysMicroCaps", funded by the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation program (Grant agreement No. ERC-2017-COG - MultiphysMicroCaps). This project is about studying microcapsules with numerical simulations and experiments in order to propose advanced models for microfluidic applications. One of its leading research axes focuses on developing new numerical approaches that provide this thesis's main setting.

Objective of the Ph.D. thesis

The objective of the thesis is to explore the use of reduced-order modeling approaches to enable fast-time simulations. This allows replacing the existing model, which despite its efficiency, it suffers from the expensive cost in terms of computation time to simulate the deformability behavior of capsules flowing in a microfluidic channel. Therefore, we are interested in investigating whether the Reduced Order Model (ROM) would be an efficient alternative to the high fidelity approach. The Ph.D. thesis thus serves the following purposes:

- The first one is proposed for a space-time-parameter mechanical problem involving spatial fields, timeline, and design variables. A Proper Orthogonal Decomposition (POD) strategy of the capsule-fluids interactions is elaborated, so it provides the time-evolution prediction of the capsule shape for any parameter values. The efficiency of POD has been proven in building reduced-order models (ROM) of microcapsules (Quesada *et al.*, 2021), but so far, no model capable of predicting capsule dynamics currently exists hence the particularity of our work. The post-treatment of the capsule shape allows deducing all the quantities of interest such as viscous load, internal tensions within the membrane, membrane energy, etc. Tracking the capsule node position field is obtained by integrating the velocity field over time. Nevertheless, the correlation of the two fields needs special care, where we deal with it using the principal modes of both fields obtained by POD decomposition through the Diffuse Approximation (DA) method and manifold learning strategy (Breitkopf *et al.*, 2002; Savignat, 2000; Raghavan *et al.*, 2013).

- The second approach expresses the FSI problem by identifying a low order nonlinear differential dynamic system. The latter describes the mechanical equilibrium law and the evolution of capsule shape. POD is used to determine reduced variables from the position and velocity collected data from FOM simulations. DMD-based model is then used to build a dynamic data-driven model verifying the kinematics and membrane equilibrium equations.

- Build a software tool that allows users to use the new model and simulate in 3D the time-evolution of the capsule shape in a microfluidic channel for any values of the parameter couple $(Ca, a/\ell)$ in few seconds.

Outline

The present manuscript is organized into four chapters. The next chapter presents the state-of-art and it is divided into two main parts. The first part defines microcapsules and gives a synthesis of their main interests. The second part consists of presenting an overview of the various modeling approaches that have been used to solve the capsule dynamics in a microfluidic channel in the literature.

In the chapter 3, we present the methods that are used in this work. That includes the problem assumptions, the summary of the integral boundary formulation and the membrane mechanics, the full order model solution, and the methods and tools used in constructing the new model.

In chapter 4, we construct a data-driven reduced-order model coupled with a manifold learning approach. The model in divided into two phases: offline and online phases. In **the offline**

phase, we construct a couple of parametric and spatial reduced-order models for displacement and velocity of the capsule membrane using a set of pre-computed simulations. These two reduced order model couples are then used to build reduced-order bases that describe the capsule displacement and velocity of any parameter vector of Ca and a/ℓ values at a selection of snapshots. In **the online phase**, we use a Diffuse Approximation procedure to estimate the principal components vector that corresponds to any admissible parameter vector of Ca and a/ℓ values. This allows building displacement and velocity snapshot matrices that are then used in manifold learning procedure to predict the deformation of the capsule in the flow for a chosen time discretization. The efficiency of the model is studied and validated numerically. A software tool is created based on the model in order to give the possibility to predict the time-evolution of capsule deformation in a microfluidic channel for any set of parameters from an admissible parametric space. Moreover, it provides visualizations of the capsule deformation and its corresponding dimensional characteristics. The model has been published in the Entropy journal (Boubehziz *et al.*, 2021).

In chapter 5, we present a dynamic reduced model using the DMD approach. In this model, we considered a reduced-order model of microcapsule dynamics under the form of a reduced dynamic system. In this latter, the motion of the capsule deformation is expressed through a Koopman operator that is identified using a dynamic mode decomposition approach. Numerical validations of the model are conducted to prove its accuracy and efficiency by comparing the obtained results to the full-order ones. Lastly, in conclusion, we summarize the contributions in these works and we discuss the perspectives opened by the thesis.

Chapter 2

Overview on capsule in flow modeling

Contents

In this chapter, we start by reviewing the concept of microcapsules and their fields of application. In section 2, we give insight into the experimental studies and the important role of numerical modelling when studying microcapsules. In section 2.2, we present the state-of-art of the used numerical models to deal with capsule in a microfluidic channel. Finally, in section 3, we give the state of art of reduced order modeling and dynamic mode decomposition that we will used in the construction of new numerical models.

1 Encapsulation: definition and applications

Encapsulation technology consists of enclosing fluid droplets in a thin elastic membrane that controls exchanges between the external and the internal of a capsule. This latter ranges from 1 μm to 1000 μm, and it is inspired from nature, where it can be found in the form of cells, bacterias, or eggs. For instance, red blood cells are an example of natural capsules with a lipid bilayer membrane supported by an actin cytoskeleton that provides wall elastic properties. It protects the hemoglobin, which is the protein responsible for transporting oxygen and the control of gas transfer to and from the tissue through the capillaries with $4-5$ μm in diameter. In industrial applications, the capsules are composed of an inner core that may take different forms (solid, liquid, and solid particle) (Singh *et al.*, 2010) and a thin elastic membrane made of a solid material that protects the inner core from the external environment.

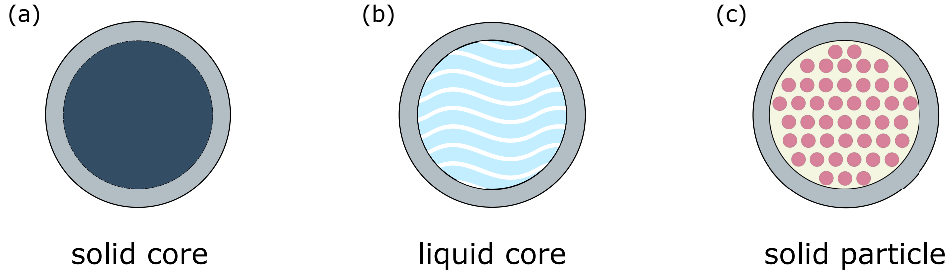


Figure 1.1: Illustration of microcapsules with a solid core (a), a liquid core (b), and solid microparticle (c).

We will restrict this study to the case of capsules with an inner liquid core. This latter has to be distinguished from vesicles and lipid bilayer membrane as red blood cells. Vesicles deform under constant volume and surface. The deformation is made possible at a low energy cost by the excess area of the membrane. Capsules, however, deform under a constant volume with their surface varying as a function of the membrane resistance to shear and area dilation (Abreu *et al.*, 2004). One of the earliest use of capsules in the industrial field appeared in the 1930s to copy typewritten material. Since then, encapsulation techniques have gained huge interest in many industrial domains such as printing (She *et al.*, 2012), biotechnology (Popel & Johnson, 2005), pharmaceutics, food industries to preserve the nutritional components and healthiness of ingredients, control their release, and mask/preserve flavors (Yang *et al.*, 2014; Nazzaro *et al.*, 2012).

One of the high potential applications of this technology is the use of liquid-core microcapsules for active-substance targeting.

This technique provides a promising solution to many biotechnological challenges (Fernandes & Gracias, 2012; Onwulata, 2012) such as neurological diseases, cancer therapy (Demirgöz *et al.*, 2009), and ischemic myocardial tissue (Ma & Su, 2013; Orive *et al.*, 2014) with effenciency. Hence, increasing therapeutic efficiency, bioavailability, and minimizing environmental damages (Ghirardi *et al.*, 1977). However, scientific challenges at this point remain to be met, such as controlling the moment of releasing and finding the optimal compromise between payload and

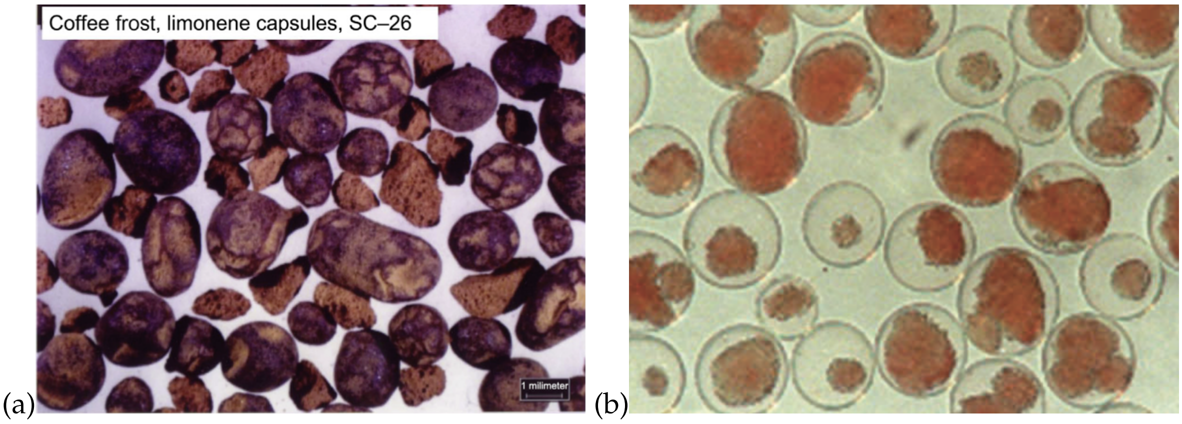


Figure 1.2: Examples of artificial microcapsules: (a) Coffee capsules containing droplets of carrier-free liquid aromas (Sobel, 2014). (b) Islet encapsulated in alginate microcapsules for Bio-artificial pancreatic (Ma & Su, 2013).

membrane resistance. Furthermore, when the capsules are injected into the flow, they are exposed to hydrodynamic loading forces that induce large membrane deformations.

2 Modeling the capsule dynamics in channel flow

Since the early 1990s, the trend toward miniaturization has made it possible to reduce the size of systems such as mechanical and fluidic systems on a micrometric scale. The design of small factor fluidic systems has allowed to study various applications in the biomedical field (Jiang *et al.*, 2016; Sandoval & Tobias, 2021; Di Natale *et al.*, 2021). The dynamics of a liquid-filled capsule in a microfluidic channel is one of the topics that have been the subject of many studies (Kuriakose & Dimitrakopoulos, 2011; Hu *et al.*, 2012; Lefebvre *et al.*, 2008). Understanding the membrane mechanical properties and flow strength on the capsule dynamics is done by studying them either experimentally or numerically. A comparison of experimental and numerical results allows to determine the membrane mechanical properties of the liquid-filled capsule.

We summarize experimental and numerical studies of capsule flow in a straight channel.

2.1 Experimental observations

When the capsule flows through the channel, it is exposed to hydrodynamic loading forces that induce large membrane deformations. This latter is characterized by a larger convex membrane curvature at the front and a concave curvature at the rear. The shapes of the deformed capsules are commonly called the parachute shape. To determine the membrane properties several techniques have been developed such as static compression and shear flow fields (Carin *et al.*, 2003; Chang & Olbricht, 1993; Pieper *et al.*, 2005), flow in microfluidic channels, micropipette (Hsu *et al.*, 2004; Heinrich & Rawicz, 2005) and atomic force microscopy (Rugar & Hansma, 1990).

The characterization of the membrane properties is done through several methods. For instance, Lefebvre *et al.* (2008) proposed observing the deformation of capsules in flow into a microfluidic channel of comparable dimensions. The observations are done as a function of the flow rate, and by using an inverse method based on a numerical model of the flowing capsule, They could deduce the membrane elastic modulus of the membrane.

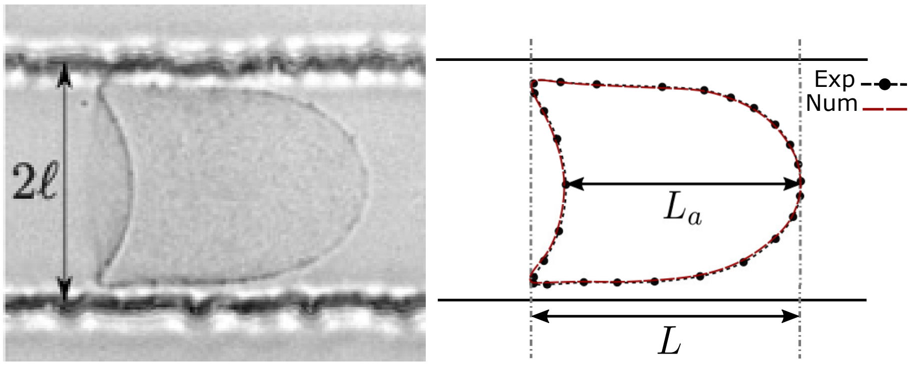


Figure 2.1: Capsule profile extraction from an experimental image: (a) experimental image; (b) profile extraction compared to simulated profile (Hu *et al.*, 2013).

De Loubens *et al.* (2014) used an elongational flow to characterize cross-linked serum albumin microcapsules. The elastics shear modulus G_s was derived from direct measurement of the deformation in function of size and albumin concentration. However, due to the limitation of the method to small deformations, it was not possible to conclude on the mechanical behavior of the membrane. The phenomenon has been widely observed in various experiments (Chu *et al.*, 2011) and is explained by the constant internal pressure at the equilibrium state, which forces the capsule profile to take a more significant curvature in the front than in the rear to fit with the viscous pressure drop of the external flow. The parachute form of the capsule is accompanied by foldings on the membrane (Cerda & Mahadevan, 2003; Luo & Pozrikidis, 2007).

2.2 Numerical models

The numerical simulations allow quantifying measurements that are usually inaccessible with experimental studies. In the literature, the motion of capsule deformability in flow is generally treated as 2D and 3D problems. The two-dimensional resolutions are simplified models that treat the problem for reasonable computation cost. Although the 2D models can provide accurate results of several aspects of the motion of a capsule, it remains unable to give a complete understanding of the problem as the 3D models (Secomb & Skalak, 1982).

The three-dimensional model was suggested for the first time by Pozrikidis (2005) in the case of modeling capsule dynamics in a simple shear flow. Since then, many variants have then been proposed to simulate solid-fluid problems. They have a commonly based on the coupling solid and fluid problems (Li *et al.*, 1988) but they differ in the way how to resolve the solid and fluid problems.

2.2.1 Numerical modeling of the wall Mechanics

Solving the solid mechanics of the capsule membrane with a suitable model is a crucial part of the overall accuracy resolution. Since the membrane thickness is considered infinitely thin, several two-dimensional constitutive laws have been proposed to model the mechanical behavior of the membrane. To find the viscous load exerted by the fluids on the membrane, two approaches are frequently used.

- **The strong form of membrane equilibrium** is solved locally at each point of the capsule wall using several approaches. Kessler *et al.* (2008) used the spectral method, which is based on the projection of the solid domain on the basis of spectral functions. The method appears to be effective, but has so far only been used in areas of small strain. (Ramanujan & Pozrikidis, 1998; Pozrikidis, 2005) treat the motion of elastic capsule membrane in a cylindrical tube, where the capsule membrane mechanics is coupled with the interior and exterior hydrodynamics by means of surface equilibrium expressed in global Cartesian coordinates, and the flow of capsule is resolved with a boundary element method. Lac *et al.* (2004) studied the case of a filled-liquid capsule suspended in an unbounded shear using the boundary-element method is used in conjunction with surface interpolation through local bicubic B-splines.

- **The weak form of membrane equilibrium** where a Finite Elements Method (FEM) is then used to find the viscous load exerted by the fluids on the membrane. Doddi & Bagchi (2008) coupled FEM through Immersed boundary method (IBM) with finite difference method. Sui *et al.* (2008) used a numerical method for large deformation of capsules based on a mixed finite-difference Fourier transform method for the flow solver and a front-tracking method for deformable interfaces. Hu *et al.* (2012) used the Boundary Integral Method (BIM) to solve the fluid problem.

The membrane deformation has been studied in various configurations, such as simple shear flow (Lac *et al.*, 2004; Li & Sarkar, 2008; Kessler *et al.*, 2008; Pozrikidis, 1995; Ramanujan & Pozrikidis, 1998), Poiseuille tube flow (Doddi & Bagchi, 2008; Lefebvre & Barthès-Biesel, 2007; Doddi & Bagchi, 2008), elongational flow (Diaz *et al.*, 2000; Dodson & Dimitrakopoulus, 2009), effect of the membrane law on the capsule deformation (Hu *et al.*, 2013), and taking into account the membrane bending stiffness (Kwak & Pozrikidis, 2001).

2.2.2 Numerical modeling of fluid mechanics

The fluid velocity field is governed by the Stokes equations and it can be solved through several approaches.

- **The Boundary Integral Method (BIM)** The evaluation of the velocity field is done on the boundaries of the fluid domain with 2D integrals (Ladyzhenskaya, 1969). This approach has been used by Pozrikidis (2005) including the capsule suspended in a simple shear flow. It allows to express the velocity of the discretized membrane as a function of an integral on the problem boundary. The discretization is applied only to the membrane, so that no remeshing of the fluid domains is needed. The method has been largely used to study the deformation and motion of the capsules in different configurations of fluid flow: simple shear flow (Ramanujan & Pozrikidis, 1998; Walter *et al.*, 2011; Foessel *et al.*, 2011) plane hyperbolic flow (Lac *et al.*, 2004; Dodson & Dimitrakopoulos, 2008; Walter *et al.*, 2010; Hu *et al.*, 2012).

- **The Lattice-Boltzmann method (LBM)** is a kinetic based approach for simulating fluid flow. It considers the fluid domains as fictive particles that interact according to the applied forces. It has been proved to be an efficient efficient method for solving complex fluid systems (Chen & Doolen, 1998). This allows treating complex boundaries, incorporating microscopic interactions.

2.2.3 Application to capsule flowing into a straight channel

In the literature, several studies are done coupling the above-presented methods to deal with the deformation of a capsule flowing into a channel. For instance, Doddi & Bagchi (2008) studied the lateral migration of a capsule in a plane Poiseuille by solving the fluid problem with the FE method coupled through Immersed Boundary Method (IBM) with finite difference method. Hu *et al.* (2012); Kuriakose & Dimitrakopoulos (2011) investigated the motion of an elastic capsule motion in the cases of square microfluidic channel and a cylindrical tube. These studies concluded that the confinement involves a large deformation of the capsule, where the capsule-dynamics are governed by the flow strength and the size ratio between the capsule and the channel. The deformation of the capsule is presented by an inversion of the back curvature, usually called parachute shape and bending. In addition, it found that capsule deformation can reach a steady-state according to Ca limitation values due to the NEO-Hookean law. FSI numerical resolutions variants are generally reliables to deal with capsule deformation (Sévénié *et al.*, 2015), whereas some of them suffer from instability and limitations issues, such as the model given by Pozrikidis (2005) where it shows numerical instability when negative tensions have occurred, Even for effective FOMs as the model that is given by Sévénié *et al.* (2015), the resolution is expensive in computation time. This can be a significant limitation, for instance, when identifying the membrane mechanical properties from experimentations. To overcome this constraint, some surrogate models based on reduced-order modeling appeared as a promising alternative that has been proven as efficient in reducing computation costs.

3 Reduced Order Modeling (ROM)

The simulation of the capsule motion in a microfluidic channel using the FSI model entails expensive in time and computation resources (Cottet *et al.*, 2008). An alternative technique, so-called reduced-order model (ROM), is used to reduce the dimensionality of the numerical model while preserving the system's main characteristics. It consists in building a reduced set of orthogonal bases that retains main variations in a large set of variables. The main interest of using a ROM is to provide a solution as close as possible to the full-order model (FOM) (Benjamin & Karen, 2015, 2016) solution with a low cost in time and computational resources. Several approaches exist to perform model reduction can be categorized according to two classifications: a priori/a posterior and intrusive/non-intrusive. A posteriori methods build a Reduced Order Model (ROM) based on a previously collected database, whereas a priori methods build prior data by relying on the knowledge of the governing equations and numerical methods used in the model. Non-intrusive methods use the simulation model as a black box, while intrusive methods involve changes to the simulation model.

One of the most popular variants of non-intrusive ROMs is meta-modeling, in which collected simulation results are used as a design of experiment to fit a surrogate model (Kriging (Cressie, 1988), Neural Network (NN) (Papadrakakis *et al.*, 1998), Non-intrusive Proper Orthogonal Decomposition (NiPOD) (Iuliano & Quagliarella, 2013; Xiao *et al.*, 2015)). Thus, meta-modeling methods are posterior non-intrusive methods. On the flip side, the Projected Reduced Order Model (PROM) is the intrusive method that approximates the unknown as a linear combination of Reduced Basis (RB) vectors. This kind of method is usually coupled with a method that approximates the projection of internal variables such as the Discrete Empirical Interpolation Method (DEIM) (Chaturantabut & Sorensen, 2010) and the Hyper-Reduction (HR).

In the model mentioned above, order reduction methods are applicable to fixed-domain problems, which is not the case for the FSI model, where this latter also needs to derive the ROM. The data-driven and non-intrusive reduced-order models (ROM) can be seen as supervised ANN (Pawar *et al.*, 2019; Xiao *et al.*, 2017). For parametrized partial differential problems, ROMs usually perform a dimensionality reduction through suitable reduced bases. This can be achieved via different approaches such as the Proper Orthogonal Decomposition (POD) (Cordier, 2008; Benner *et al.*, 2015; Silva & Alvaro, 2015), piecewise tangential interpolation (Gallivan *et al.*, 2004), Proper Generalized Decompositions (PGD) (Chinesta *et al.*, 2011, 2010; Ghnatios *et al.*, 2012), Empirical Interpolation Methods (EIM) (Maxime *et al.*, 2004; Chaturantabut & Sorensen, 2010; Xiao *et al.*, 2014) or via different greedy procedures (Lappano *et al.*, 2016). Then one has to find the manifold that maps the parameters to the coefficients of a linear combination of the reduced basis functions. For example, this can also be supervised using universal approximation techniques like diffuse approximation (Breitkopf *et al.*, 2002). Using ROMs may lead to substantial speedups as compared to FOM.

3.1 Proper Orthogonal Decomposition (POD)

The Proper Orthogonal Decomposition (POD) method is a statistical data analysis tool introduced by Lumley (1967) for analyzing turbulent flows. He showed that the principal components corresponded to some coherent spatial structures that occurred cyclically in time.

Later, with the introduction of the Galerkin approximation, the idea to approximate the solution in the space generated by a finite number of principal components came to light. For a few decades now, researchers have developed several variants of this approach. For instance, Sirovich (1987) suggested an alternative way to construct POD modes using snapshots collected from numerical or empirical data. Aubry *et al.* (1988) gave a first model that could reproduce realistic low-dimensional ODE systems for a turbulent flow model. Then Aubry (1991) proposed a deterministic space-time symmetric version of the POD by using a bi-orthogonal decomposition that expands orthogonal modes in time and space. Christensen *et al.* (1999) suggested relying on snapshots with weights to obtain more relevant modes. Sieber *et al.* (2016) proposed involving an additional temporal constraint that enables a clear separation of phenomena that occur at multiple frequencies and energies. In the following, we give interest to the snapshot-POD variant, which is based on building an orthogonal basis by extracting relevant information from pre-computed snapshots. This latter is usually collected from specific numerical or experimental observations and gathered in a snapshot matrix. The POD basis is obtained by performing a singular value decomposition (SVD) involving a reduced number of snapshots (Nayroles *et al.*, 1992).

3.2 Dynamic Method Decomposition (DMD)

DMD was introduced in the fluid mechanic community by Schmid (2010) as a generalization of linear stability analysis, and it can handle nonlinear systems. Since then, it has quickly gained popularity and has emerged as a powerful alternative tool to POD for analyzing the dynamics. While POD modes reconstruct a dataset with the modes ranked in terms of energy content, the DMD modes and eigenvalues describe the dynamics observed in the time series in terms of oscillatory components Schmid (2010). We also refer to Rowley *et al.* (2009) DMD variant, where the decomposition is done by considering DMD modes as a subset of Koopman modes. Furthermore, the DMD model has gained a growing interest that is expressed through multiple variants of the method. For instance, Chen *et al.* (2012) proposed an optimized DMD that can better identify characteristic flow frequencies in some cases. Williams *et al.* (2015) gave an improved version of DMD called the Extended DMD (EDMD) method, in which the Koopman decomposition is approximated using nonlinear observables that spans a subspace of the space of scalar observables. However, this variant suffers from the fast increase in computational cost. To circumvent this issue, kernel DMD (KDMD) (Kevrekidis *et al.*, 2016) proposed as an alternative, which uses a kernel function to implicitly include nonlinear snapshots set while keeping the computational cost of standard DMD. Héas & Herzet (2017) suggested an improved version of KDMD called optimal KDMD, in which Noack *et al.* (2016) developed a recursive dynamic mode decomposition (RDMD) that combines POD with DMD. Le Clainche & Vega (2017) proposed an extension called High order DMD where Erichson *et al.* (2019) suggested randomized DMD where a low-rank matrix is computed from a high dimensional matrix.

DMD has been used in several examples. We mention the work of Muld *et al.* (2012), where they applied DMD to the flow around high-speed trains. Wu & Martin (2007) treated shock-turbulent boundary layer interactions. Seena & Sung (2011) used DMD to identify the large-scale vortical structures responsible for hydrodynamic oscillations to the pressure fluctuations of incompressible turbulent flows over an open cavity.

4 Conclusion

This chapter gave an overview of the definition of a microcapsule and the main interest application of this technology. We have also reviewed different models to model the deformation of a liquid-filled capsule in flow numerically. We could notice that the most reliable three-dimensional models that simulate capsule deformability in flow, are so expensive in term of computation time. The reduced-order model approach and the reduced dynamic system can be effective alternatives to the 3D high-fidelity models. Especially since Quesada *et al.* (2021) have proved the effectiveness of the reduced-order model approach to simulate capsule deformability in a microfluidic channel at the state-state. In the next chapter, we will give the problem statement and the main ingredients to build two ROMs that can handle the time-evolution problem of capsule deformation in a microchannel.

Chapter 3

Material and methods

Contents

This chapter introduces the mathematical model of a spherical capsule flowing in a square channel. We start section 1 by giving the problem statement describing the behavior of a capsule in the channel based on a fluid-structure interaction model proposed by (Hu *et al.*, 2012). The resolution is done by coupling two problems: A solid problem that describes the capsule membrane deformation through a variational form and a fluid problem in which the internal and external flows are described using the boundary integral method (BIM). Then, we highlight in section 2 the general numerical schemes of Fluid-Structure Interaction (FSI) to simulate the capsule motion in a square channel. Furthermore, we also present the main tools for the construction of the new fast time simulation models as Proper Orthogonal Decomposition (POD), Dynamic Mode Decomposition (DMD), and some manifold learning. We finish the chapter by giving the accuracy criteria that will be used to validate the new model.

1 Problem statement

Let us consider a spherical capsule of radius a freely placed in a three-dimensional channel with a square cross-section of length 2ℓ (see Figure 1.1). The choice of using a square-section channel rather than a cylindrical one is based on the idea to compare numerical with experimental results where microfluidic channels are created using soft-lithography technique, which is the same as the one used to print electronic circuits. The capsule and the channel are filled with an incompressible Newtonian fluid of the same constant density ρ and dynamic viscosity μ. The capsule is enclosed by a thin hyperelastic isotropic membrane (surface shear modulus G_s, area expansion modulus $K_s = 3G_s$). It is subjected to a Poiseuille flow of mean velocity V.

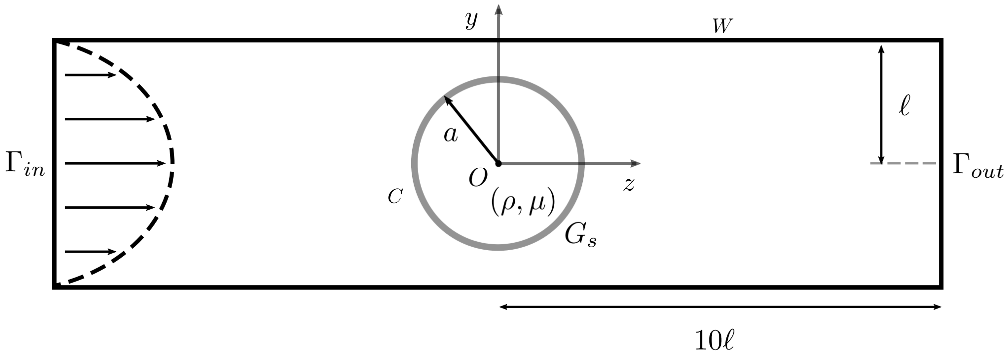


Figure 1.1: Initial configuration of a spherical capsule flowing in a square channel.

The problem is governed by two dimensionless numbers:

- The confinement ratio a/ℓ, ratio of the capsule to tube sizes;
- The capillary number $Ca = \mu V/G_s$, ratio of the viscous forces onto the capsule membrane to the membrane elastic forces.

The flow Reynolds number is assumed to be very small, inertia being negligible. Hence, the flow in the internal ($\beta = in$) and external ($\beta = ex$) fluids are described by the Stokes equations:

$$\nabla \cdot \boldsymbol{v}^{\beta} = 0,\ \nabla \cdot \boldsymbol{\sigma}^{\beta} = 0,\ \beta = in, ex. \tag{1.1}$$

where $\boldsymbol{\sigma}^{\beta}$ is the stress tensor in the fluids. The Stokes equations are defined in the domains bounded by the capsule membrane for $\beta = in$ and the channel wall for $\beta = ex$. The inlet Γ_{in} and outlet Γ_{out} cross-sections of the channel are assumed to be far from the capsule (10ℓ). The reference frame (O, x, y, z) is fixed on the capsule center of mass O at each time step. For the velocity vector field $\boldsymbol{v}^{\beta}$ and the pressure field p^{β}, we consider the following boundary conditions:

- The flow perturbation induced by the capsule vanishes at Γ_{in} and Γ_{out}:

  $$\boldsymbol{v}^{ex}(\boldsymbol{x}, t) \to \boldsymbol{v}^{\infty}(\boldsymbol{x}), \text{ when } \boldsymbol{x} \in \Gamma_{in} \cup \Gamma_{out}, \tag{1.2}$$

  where $\boldsymbol{v}^{\infty}$ is the flow velocity of the suspending fluid in the absence of capsule.

- Uniform pressure at Γ_{in} and Γ_{out}:

  $$p^{ex}(\boldsymbol{x}, t) = 0, \ \boldsymbol{x} \in \Gamma_{in}, \tag{1.3}$$

  $$p^{ex}(\boldsymbol{x}, t) = \Delta p(t) + \Delta p^{\infty}, \ \boldsymbol{x} \in \Gamma_{out}, \tag{1.4}$$

  where Δp^{∞} is the undisturbed suspending pressure drop in the absence of capsule and Δp is the additional pressure drop due to the capsule.

- No slip on the channel wall W:

  $$\boldsymbol{v}^{ex}(\boldsymbol{x}, t) = 0, \text{ for } \boldsymbol{x} \in W \tag{1.5}$$

- No slip on the capsule membrane C:

  $$\boldsymbol{v}^{in}(\boldsymbol{x}, t) = \boldsymbol{v}^{ex}(\boldsymbol{x}, t) = \frac{\partial}{\partial t}\boldsymbol{x}(\boldsymbol{X}, t), \text{ for } \boldsymbol{x} \in C, \tag{1.6}$$

  where $\frac{\partial}{\partial t}\boldsymbol{x}(\boldsymbol{X}, t)$ is the membrane velocity at position $\boldsymbol{x}$ at time t, and $\boldsymbol{X}$ is the reference position vector of the capsule membrane.

- The normal loading continuity indicates that the load $\boldsymbol{q}$ on the membrane is due to the viscous traction jump

  $$\left[\boldsymbol{\sigma}^{ex}(\boldsymbol{x}) - \boldsymbol{\sigma}^{in}(\boldsymbol{x})\right] \cdot \boldsymbol{n} = \boldsymbol{q}, \text{ for } \ \boldsymbol{x} \in C, \tag{1.7}$$

  where $\boldsymbol{n}$ is the outward unit normal vector.

As the membrane thickness is negligibly small compared to the capsule dimensions, the membrane can be considered as a hyperelastic surface devoid of bending stiffness. The in-plane deformation is then measured by the principal extension ratios λ_1 and λ_2, which measure the in-plane deformation. The capsule can be highly deformed due to the combined effects of hydrodynamic forces, boundary confinement, and membrane deformability. Consequently, the choice of membrane constitutive law is essential. We consider the Neo-Hookean (NH) constitutive law that models the membrane as an infinitely thin sheet of a three-dimensional isotropic and incompressible material. It was indeed shown to adequately model microcapsules with a cross-linked protein membrane (Chu *et al.*, 2011; Hu *et al.*, 2013; Wang *et al.*, 2021). The principal Cauchy in-plane tensions τ_i $(i = 1, 2)$ (forces per unit arc length of deformed surface curves) can be expressed as a function of the principal extension ratios:

$$\tau_1 = \frac{G_s}{\lambda_1 \lambda_2} \left[\lambda_1^2 - \frac{1}{(\lambda_1 \lambda_2)^2} \right] \qquad \text{(likewise for } \tau_2\text{)}. \tag{1.8}$$

1.1 Solid problem: Membrane mechanics

The capsule motion is a complex fluid-structure interaction problem. We highlight in this section the most relevant fundamental concepts such as the description of membrane deformation, the constitutive laws, and membrane equilibrium (Walter *et al.*, 2010).

1.1.1 Membrane deformation

The thin membrane of the capsule is considered in the problem as an impermeable hyperelastic isotropic elastic surface with surface shear elastic modulus G and area dilatation modulus K. The membrane deformation is determined through the local covariant and contravariant bases that are constructed from the surface curvilinear coordinates (Green *et al.*, 1971; Walter, 2009). Let $\boldsymbol{X}$ the position of a membrane material point in the reference state, and $\boldsymbol{x}(\boldsymbol{X}, t)$ is its displacement position in the deformed state. The local deformation of the membrane surface can be measured by the Green-Lagrange strain tensor

$$e = \frac{1}{2} \left(\boldsymbol{F}^T \cdot \boldsymbol{F} - \boldsymbol{I} \right), \tag{1.9}$$

where $\boldsymbol{I}$ is the identity tensor and $\boldsymbol{F} = \frac{\partial x}{\partial X}$ is the gradient transformation. The membrane deformation can be determined by the principal dilatation ratios λ_1, and λ_2 in its plane, which correspond to two eigenvalues of e. The invariants of the transformation are given

$$I_1 = \lambda_1^2 + \lambda_2^2 - 2, \; I_2 = (\lambda_1 \lambda_2)^2 - 1 = \boldsymbol{J}_s^2 - 1, \tag{1.10}$$

where the Jacobian $J_s = det(\boldsymbol{F}) = \lambda_1\lambda_2$ measures the ratio of deformed membrane area to the reference area.

1.2 Membrane constitutive laws

The material mechanical properties are given by the strain energy function $w_s(I_{s1}, I_{s2})$ per unit area of the undeformed membrane. Since the capsule membrane is infinitely thin, the elastic stresses in the membrane are replaced by Cauchy tension tensors $\hat{\boldsymbol{T}}$, corresponding to forces per unit arc length measured in the plane of the membrane deformation. The Cauchy tension tensor $\boldsymbol{T}$ is related to a strain energy function to $w_s\,(I_1, I_2)$ per unit deformed surface area (Green & Adkins, 1970; Walter, 2009):

$$\hat{\boldsymbol{T}} = \frac{1}{\boldsymbol{J}_s}\boldsymbol{F}_s \cdot \frac{\partial w_s}{\partial e_s} \cdot \boldsymbol{F}_s^T, \tag{1.11}$$

where $\frac{\partial w_s}{\partial e_s}$ is the Piola–Kirchhoff tension tensor.

The membrane constitutive law is defined by the relation between the membrane stresses $\hat{\boldsymbol{T}}$ and the two principal extension ratios in the membrane plane λ_1 and λ_2. According to the material membrane behavior, an adapted constitutive law can be used to model capsule membrane (Barthès-Biesel *et al.*, 2002). There are two most commonly used membrane constitutive laws to describe the mechanics of a thin membrane: The strain-softening neo-Hookean law and the strain-hardening Skalak law.

1.2.1 The neo-Hookean law

This law follows the assumption that the membrane is an infinitely thin sheet of a three-dimensional isotropic and incompressible material. It is appropriate to model the behavior of protein-reticulated membrane (Carin *et al.*, 2003; Chu *et al.*, 2011) where for a two-dimensional material, the strain energy function of the neo-Hookean law is given by

$$w_s^{NH} = \frac{G_s^{NH}}{2}\left(I_{s1} - 1 + \frac{1}{I_{s2}+1}\right). \tag{1.12}$$

The area dilatation is balanced by membrane thinning because of the volume incompressibility. The link between the area dilatation modulus and the surface shear modulus is $K_s^{NH} = 3G_s^{NH}$ when $\nu_s = 0.5$.

1.2.2 Membrane equilibrium

Under the assumption of an infinitely thin membrane, the capsule membrane is considered an impermeable hyperelastic isotropic surface with surface shear elastic modulus G_s and area dilatation modulus K_s. The membrane behavior is governed by the local equilibrium equation

$$\nabla_s \cdot \boldsymbol{T} + \boldsymbol{q} = 0, \tag{1.13}$$

where ∇_s is the surface divergence operator in the deformed configuration and $\boldsymbol{q}$ is the external load on the membrane.

The weak form of the equation can be written according to the principle of virtual work, where the virtual work of external and internal fluid forces are balanced for any virtual displacement field $\hat{\boldsymbol{u}}$ (Walter, 2009).

$$\int_{S(t)} \hat{\boldsymbol{u}} \cdot \boldsymbol{q} dS + \int_{S(t)} \hat{\epsilon}\left(\hat{\boldsymbol{u}}\right) : \boldsymbol{T} dS = 0, \tag{1.14}$$

where $\hat{\epsilon}\left(\hat{\boldsymbol{u}}\right) = \frac{1}{2}\left(\nabla_s \hat{\boldsymbol{u}} + \nabla_s \hat{\boldsymbol{u}}^T\right)$ denotes the virtual deformation tensor and $\hat{\boldsymbol{u}}$ a virtual deformation. The first term corresponds to the virtual work of the external fluid forces, while the second side corresponds to the virtual work of the membrane elastic forces.

1.3 Fluid problem: Boundary Integral formulation for confined capsule flow

In order to give the fluid formulation of the three-dimensional motion of the internal and external fluids, we consider the capsule placed in a microfluidic channel to be composed of two parts (Pozrikidis, 2005): the undisturbed channel flow in the absence of the capsule, and the disturbed channel flow in the presence of the capsule. Hence the velocity $\boldsymbol{v}(\boldsymbol{x})$ at any point $\boldsymbol{x}$ in the fluid domain of a confined capsule flow is given by:

$$\boldsymbol{v}\left(\boldsymbol{x}\right) = \boldsymbol{v}^{\infty}\left(\boldsymbol{x}\right) - \frac{1}{8\pi\mu}\left[\int_C \tilde{\boldsymbol{J}}\left(\boldsymbol{r}\right) \cdot q dS\left(\boldsymbol{y}\right) + \int_W \tilde{\boldsymbol{J}}\left(\boldsymbol{r}\right) \cdot \tilde{f} dS\left(\boldsymbol{y}\right) - \Delta P \int_{\Gamma_{out}} \tilde{\boldsymbol{J}}\left(\boldsymbol{r}\right) \cdot \boldsymbol{n} dS\left(\boldsymbol{y}\right)\right] \tag{1.15}$$

where $\tilde{f}$ is the disturbance wall friction due to the capsule presence, $\boldsymbol{v}^{\infty}$ denotes the undisturbed flow in the absence of capsule, $\tilde{\boldsymbol{J}}$ and $\boldsymbol{K}$ denote the Green kernels defined by

$$\tilde{\boldsymbol{J}}(\boldsymbol{r}) = \frac{1}{\boldsymbol{r}}\mathbb{I} + \frac{\boldsymbol{r} \otimes \boldsymbol{r}}{\boldsymbol{r}^3}, \qquad \boldsymbol{K} = -6\,\frac{\boldsymbol{r} \otimes \boldsymbol{r} \otimes \boldsymbol{r}}{\boldsymbol{r}^5} \tag{1.16}$$

where $\boldsymbol{r} = \boldsymbol{y}_s - \boldsymbol{x}_s$ represents the distance of a point $\boldsymbol{y}_s$ on the capsule surface $\mathcal{S}_t$ from a point $\boldsymbol{x}_s$, where the velocity vector is computed. The uniform axial velocity distribution $\boldsymbol{v}^{\infty} = \boldsymbol{v}^{\infty}(\boldsymbol{z}, \boldsymbol{y})e$, is given by

$$\boldsymbol{v}^{\infty}\left(\boldsymbol{x}, \boldsymbol{y}\right) = \frac{\pi V \sum_n^{\infty} \frac{1}{n^3}\left[1 - \frac{\cosh n\pi \boldsymbol{x}/\ell}{\cosh\left(n\pi/2\right)}\right] \sin n\pi\left(\boldsymbol{y}/\ell + 1/2\right)}{2\left[\frac{\pi^4}{96} - \sum_n^{\infty} \frac{\tanh\left(n\pi/2\right)}{n^5\pi/2}\right]} \qquad n = 1, 2, 3, \ldots \tag{1.17}$$

Applying the reciprocal theorem to the flow with and without the capsule gives

$$\int_C \left[\boldsymbol{\sigma}^{(1)} \cdot \boldsymbol{n}\right] \cdot v^{\infty} dS - \left(\Delta P + \Delta P^{\infty}\right) Q = \int_C \left[\boldsymbol{\sigma}^{\infty} \cdot \boldsymbol{n}\right] \cdot \boldsymbol{v}^{(1)} dS - \Delta P^{\infty} Q, \tag{1.18}$$

where the flow rate Q is the same with and without the capsule, and the velocity is zero on W. By applying the reciprocal theorem to $(\boldsymbol{v}^{\infty}, \boldsymbol{\sigma}^{\infty})$ and $\left(\boldsymbol{v}^{(2)}, \boldsymbol{\sigma}^{(2)}\right)$ in the internal flow domain bounded by C, we get

$$\int_C \left[\boldsymbol{\sigma}^{(2)} \cdot \boldsymbol{n}\right] \cdot \boldsymbol{v}^{\infty} dS = \int_C \left[\boldsymbol{\sigma}^{\infty} \cdot \boldsymbol{n}\right] \cdot \boldsymbol{v}^{(2)} dS. \tag{1.19}$$

with the assumption of no-slip condition on the capsule membrane surface, we find that the additional pressure drop caused by the capsule is simply given by

$$\Delta P = \frac{1}{Q} \int_C \boldsymbol{v}^{\infty}(\boldsymbol{x}) \cdot \boldsymbol{q} dS(\boldsymbol{y}). \tag{1.20}$$

Once the value of the pressure disturbance ΔP is known, the equation 1.15 at $\boldsymbol{x} \in W$ becomes

$$0 = \int_C J(\boldsymbol{r}) \cdot \boldsymbol{q} dS(\boldsymbol{y}) + \int_W \tilde{\boldsymbol{J}}(\boldsymbol{r}) \cdot \boldsymbol{f} dS(\boldsymbol{y}) - \Delta P \int_{\Gamma_{out}} \tilde{\boldsymbol{J}}(\boldsymbol{r}) \cdot \boldsymbol{n} dS(\boldsymbol{y}) \tag{1.21}$$

which can be solved numerically.

2 Fluid-Structure Interaction (FSI) model

The Fluid-Structure Interaction (FSI) problem is solved numerically by coupling the Finite Element Method (FEM) that solves the membrane mechanical problem (Walter *et al.*, 2010; Hu *et al.*, 2012) with the Boundary Integral Method (BIM) that solves the fluid equations for the internal and external flows using the Caps3D in-house code. The unknowns are the discrete displacement field $\{\boldsymbol{u}\}$ and the discrete velocity field $\{\boldsymbol{v}\}$ at the nodes of the membrane mesh. The equation of kinematics states that $\frac{d}{dt}\{\boldsymbol{u}\} = \{\boldsymbol{v}\}$. The forces exerted onto the membrane are computed by the FEM. The deformation of the membrane is computed from the velocity vector field obtained at the membrane nodes by solving the Stokes equations with the BIM, leading to a nonlinear relation written in abstract form $\{\boldsymbol{v}\} = \varphi(\{\boldsymbol{u}\})$.

The numerical resolution is done by discretizing the capsule membrane surface by $3N_x$ nodes and N_E elements. The coupled resolution BI+FE process as follows:

1. From an initial position $\{\boldsymbol{x}\}$ $(\{\boldsymbol{X}\}, t_0)$, the weak form of the wall equilibrium equation (1.14) is solved to determine the unknown viscous load $\boldsymbol{q}$ exerted by the fluids on the wall.

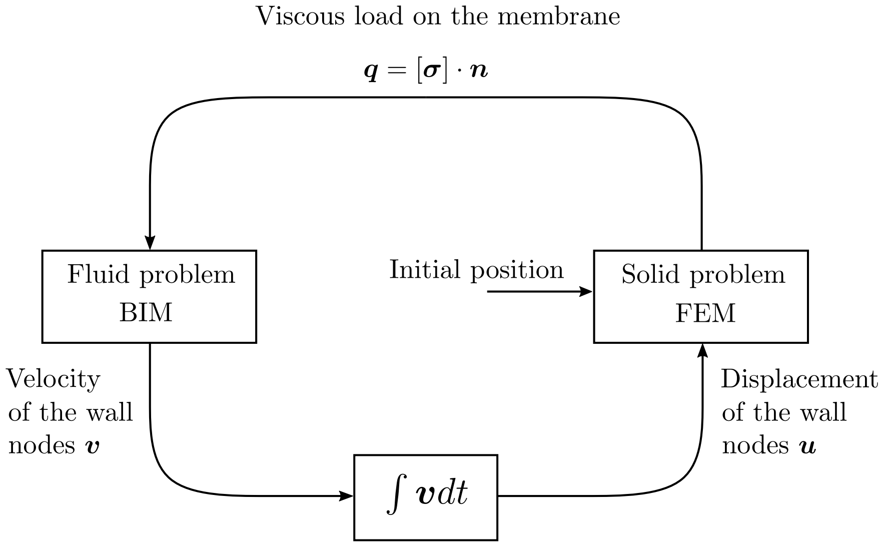


Figure 2.1: An illustration of Fluid-Solid Interaction (FSI) resolution.

2. The viscous load on the membrane $\{q\} = [\sigma] \cdot \{n\}$ is computed from the deformed shape and used to modify the fluid flows.

3. Once the viscous load $\{q\}$ is known, the fluid velocity at the nodes is obtained from the boundary integral equation (1.15).

4. The fluid velocity is integrated in time to deduce the displacement of the mesh nodes. The positions of the nodes are updated for the next loop.

2.1 Pre-deformation for large capsules $a/\ell > 0.95$

When the capsule's initial size is larger than the channel cross dimension, the capsule needs to be pre-deformed to fit inside the channel. This is be done by applying on the capsule nodes with the locations $(\{x\}, \{y\}, \{z\})$ a transformation such that

$$\{\tilde{x}\} = k_1\{x\}, \ \{\tilde{y}\} = k_1\{y\}, \ \text{ and } \ \{\tilde{z}\} = k_2\{z\}, \tag{2.1}$$

where $k_1^2 k_2 = 1$ and k_1 is chosen such that $\{y\}/\ell = 0.95$ in order to avoid contact between the membrane and the channel wall and give more stale simulations.

2.2 Discrete full order model (FOM)

The Fluid-Structure Interaction (FSI) problem is numerically modeled by coupling the Boundary Integral Method (BIM) that solves the fluid equations with the Finite Element Method (FEM) that

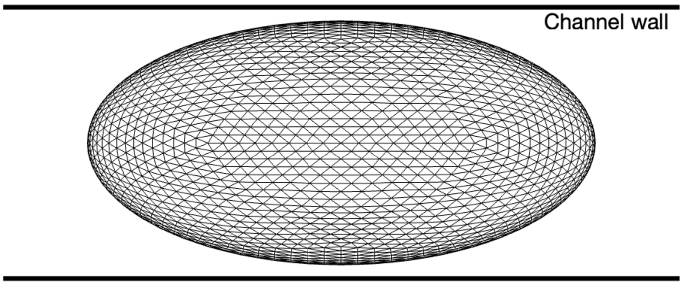


Figure 2.2: Initial spherical capsule with size ratio $a/\ell \geq 0.95$ in order to fit inside the channel.

solves the membrane mechanical problem (Walter *et al.*, 2010; Hu *et al.*, 2012) using the Caps3D in-house code. For a given parameter vector $\boldsymbol{\theta} = (\theta_1, \theta_2)^T$, where $\theta_1 = Ca$ and $\theta_2 = a/\ell$, the time-continuous semi-discrete FSI scheme reads in abstract form

$$\begin{aligned}
&\frac{d}{dt}\{\boldsymbol{u}\}(t) = \{\boldsymbol{v}\}(t),\\
&\{\boldsymbol{v}\}(t) = \{\boldsymbol{\varphi}\}(\{\boldsymbol{u}\}(t), \boldsymbol{\theta}), \quad t \in (0, T_f],\\
&\{\boldsymbol{u}\}(0) = \{\boldsymbol{0}\},\ \{\boldsymbol{v}\}(t) = \{\boldsymbol{\varphi}\}(\{\boldsymbol{0}\}, \boldsymbol{\theta})
\end{aligned}$$

where $\{\boldsymbol{u}\}(t)$ and $\{\boldsymbol{v}\}(t)$ represent the discrete FE displacement and velocity fields at continuous time t, and T_f is the final time. For time discretization, either a forward Euler scheme or a second order Runge-Kutta scheme is used with a suitable constant time step $\delta t > 0$. The Euler scheme reads

$$\begin{aligned}
&\{\boldsymbol{u}^{i+1}\} = \{\boldsymbol{u}^i\} + \delta t\,\{\boldsymbol{v}^i\},\\
&\{\boldsymbol{v}^{i+1}\} = \{\boldsymbol{\varphi}\}(\{\boldsymbol{u}^{i+1}\}, \boldsymbol{\theta}),\\
&\{\boldsymbol{u}^0\} = \{\boldsymbol{0}\},\ \{\boldsymbol{v}^0\} = \{\boldsymbol{\varphi}\}(\{\boldsymbol{0}\}, \boldsymbol{\theta})
\end{aligned}$$

where $\{\boldsymbol{u}^i\}$ and $\{\boldsymbol{v}^i\}$ represent the discrete FE displacement and velocity fields at discrete time $t^{i,\delta} = i\,\delta t \leq T_f$. For second-order accuracy in time, a Runge-Kutta Ralston scheme is used:

$$\begin{aligned}
&\{\widehat{\boldsymbol{u}}^{i+2/3}\} = \{\boldsymbol{u}^i\} + \frac{2}{3}\delta t\,\{\boldsymbol{v}^i\},\\
&\{\widehat{\boldsymbol{v}}^{i+2/3}\} = \{\boldsymbol{\varphi}\}(\{\widehat{\boldsymbol{u}}^{i+2/3}\}, \boldsymbol{\theta}),\\
&\{\boldsymbol{u}^{i+1}\} = \{\boldsymbol{u}^i\} + \frac{\delta t}{4}\left(\{\boldsymbol{v}^i\} + 3\{\widehat{\boldsymbol{v}}^{i+2/3}\}\right),\\
&\{\boldsymbol{v}^{i+1}\} = \{\boldsymbol{\varphi}\}(\{\boldsymbol{u}^{i+1}\}, \boldsymbol{\theta}),\\
&\{\boldsymbol{u}^0\} = \{\boldsymbol{0}\},\ \{\boldsymbol{v}^0\} = \{\boldsymbol{\varphi}\}(\{\boldsymbol{0}\}, \boldsymbol{\theta}).
\end{aligned}$$

Because of the explicit nature of the numerical schemes for the equation of kinematics, the time step is subject to a Courant-Friedrichs-Lewy (CFL)-like stability condition

$$\dot{\gamma}\,\delta t < C\,\frac{\Delta h_C}{\ell}\,Ca, \tag{2.2}$$

where $\dot{\gamma} = V/\ell$, $C > 0$ is a known constant and Δh_C is the typical mesh size (see (Walter *et al.*, 2010)).

Proper Orthogonal Decomposition has been shown to be particularly suitable to build reduced order models (ROM) of microcapsules (Quesada *et al.*, 2021), but so far no model capable of predicting capsule dynamics currently exists. The originality of the paper is to propose a ROM of the capsule-fluids interactions which provides the time-evolution of the capsule shape for any parameter values. From the capsule shape, it is indeed possible to deduce all the quantities of interest (viscous load, internal tensions within the membrane, membrane energy, etc) in post-treatment. The ROM is inspired from the physical problem, in which the boundary condition stipulates that the fluid velocity equals the capsule membrane velocity. The correction of the capsule node position field can thus be obtained by integrating the velocity field over time. The challenge remains to correlate the position and velocity fields, which we propose to do with diffuse approximation and manifold learning (Breitkopf *et al.*, 2002; Savignat, 2000; Raghavan *et al.*, 2013) using the principal modes of both fields obtained by POD decomposition.

3 Proper Orthogonal Decomposition

Reduced order modeling aims at deriving a lightweight model of low-order dimension from solutions obtained by the FOM, while trying to keep the same order of accuracy. There are many reasons for doing that. In particular, parameter exploration and sensitivity analysis are made easier because of large speedups using the ROM compared to the prohibitive FOM computational time. One can also imagine real-time parameter exploration and visualization of capsule evolution.

3.1 Definition

The Proper Orthogonal Decomposition is an "a posteriori" method for model order reduction. It aims to construct a low-dimensional approximate model by finding an orthonormal basis considerably smaller than the high fidelity model. This low-dimensional approximation model $f(\boldsymbol{x}, t)$ is constructed using the information extracted from previously computed simulations, and it is defined over the spatio-temporal domain of interest through a linear combination

$$f\left(\{\boldsymbol{X}\},t\right)\simeq\sum_{k=1}^{N_b}\alpha_k\left(t\right)\boldsymbol{\psi}_k\left(\{\boldsymbol{X}\}\right), \tag{3.1}$$

where $\boldsymbol{\alpha}\left(t\right)$ is the vector of temporal coefficients and $\boldsymbol{\psi}\left(\{\boldsymbol{X}\}\right)$ is an orthogonal basis. The formula 3.1 becomes exact when $N_b\rightarrow\infty$. The proper orthogonal decomposition consists in finding the orthogonal basis $\left(\boldsymbol{\psi}_k\right)_{k=1:N_b}$ where $\boldsymbol{\psi}_k\in\mathbb{R}^N_x$ and N_x is its dimension.

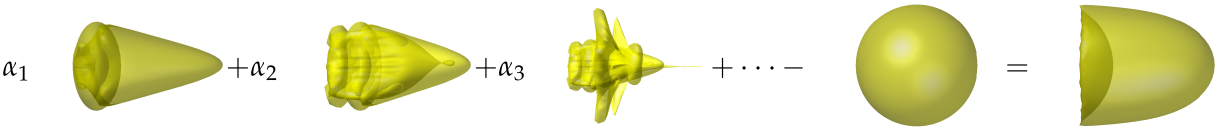


Figure 3.1: An illustration of the linear combination 3.1 between coefficients $\left(\alpha_k\right)_{k=1,\ldots,N_b}$ and modes $\left(\boldsymbol{\psi}_k\right)_{k=1,\ldots,N_b}$ at time t.

To compute the orthogonal basis $\left(\boldsymbol{\psi}_k\right)_{k=1:N_b}$, we give the snapshot matrix $A\in\mathbb{R}^{N_x\times N_t}$ as follows

$$A=\left[f\left(\{\boldsymbol{X}\},t_1\right)\quad f\left(\{\boldsymbol{X}\},t_2\right)\quad\cdots\quad f\left(\{\boldsymbol{X}\},t_{N_t}\right)\right], \tag{3.2}$$

Hence, by applying Singular Value Decomposition (SVD) (Golub & Reinsch, 1971), A can be written as

$$A=\quad U\Sigma\hat{V}^T \tag{3.3}$$

where $U\in\mathcal{M}_{N_x,N_x}\left(\mathbb{R}\right)$ and $\hat{V}\in\mathcal{M}_{N_t,N_t}\left(\mathbb{R}\right)$ are orthogonal matrices, and $\Sigma=diag\left(\sigma_1,\ldots,\sigma_r\right)\in\mathcal{M}_{N_x,N_t}\left(\mathbb{R}\right)$ with $\sigma_1\geq\sigma_2\geq\ldots\geq\sigma_r$. Here $r\leq min(N_x,N_t)$ denotes the rank of A, which is strictly smaller than N_t if the snapshot vectors are not all linearly independent. Then, we can write

$$A\boldsymbol{\psi}_i=\sigma_i\alpha_i\text{ and }A^T\alpha_i=\sigma_i\boldsymbol{\psi}_i,\qquad i=1,\ldots,r \tag{3.4}$$

or, equivalently,

$$A^TA\boldsymbol{\psi}_i=\sigma_i^2\boldsymbol{\psi}_i\text{ and }AA^T\alpha_i=\sigma_i^2\alpha_i,\qquad i=1,\ldots,r \tag{3.5}$$

i.e. σ_i^2, $i=1,\ldots,r$, are the nonzero eigenvalues of the matrix A^TA (and also of AA^T), listed in increasing order. The matrix $\hat{C}=A^TA\in\mathcal{M}_{N_t}\left(\mathbb{R}\right)$ is called correlation matrix; its elements are given by

$$\hat{C}_{ij}=f\left(\boldsymbol{x},t_i\right)^Tf\left(\boldsymbol{x},t_j\right),\qquad 1\leq i,j\leq N_t. \tag{3.6}$$

For any $N_b \leq N_t$, the POD basis $\hat{V} \in \mathcal{M}_{N_x,N_b}(\mathbb{R})$ is defined as the set of the first N_b left singular vectors of U or, equivalently, the set of vectors

$$\alpha_j = \frac{1}{\sigma_j} A \boldsymbol{\psi}_j, \qquad 1 \leq j \leq N_b, \tag{3.7}$$

obtained from the first N_b eigenvectors of the correlation matrix $\hat{C}$.

4 Dynamic Decomposition Model definition

Let us give the standard definition of DMD by assuming a sequential set of data vectors $z_1, \ldots, z_{N^t}$ in which the order of z_k $(n \times 1) - vector$ for $k = 1, \ldots, N^t$ is crucial (Schmid, 2010). We then define DMD in terms of the $n \times m$ data matrices with $m = N^t - 1$ by considering data pairs $(X;Y)$ where they are defined as follows:

$$X = [z_1, \ldots, z_{m-1}] \text{ and } Y = [z_2, \ldots, z_m] \tag{4.1}$$

The standard DMD procedure assumes that for a some (unknown) matrix A satifies:

$$A = YX^{\dagger} \tag{4.2}$$

where $X^{\dagger}$ is the pseudo-inverse of X. The dynamic mode decomposition of the pair (X, Y) is given by the eigen-decomposition of A. That is, the DMD modes and eigenvalues are the eigenvectors and eigenvalues of A.

When $n \geq m$, the above algorithm can be modified to reduce computational costs. For instance, the SVD of X in rank-r truncation can be computed efficiently.

$$X = U_r \Sigma_r V_r^T \tag{4.3}$$

So, we define the matrix $\tilde{A}$ as

$$\tilde{A} = U_r^T Y V_r \Sigma_r^{-1} \tag{4.4}$$

Compute eigenvalues and eigenvectors of $\tilde{A}$, writing $\tilde{A}W = \Lambda W$, so that Λ is a diagonal matrix containing eigenvalues λ_k and the columns of W are the eigenvectors.

Each nonzero eigenvalue λ is a DMD eigenvalue and the DMD mode corresponding to λ is then given by

$$\Phi_r = U_r^T Y V_r \Sigma_r^{-1} W \tag{4.5}$$

When $n \geq m$, the above algorithm can be modified to reduce computational costs. For instance, the SVD of X can be computed efficiently using the method of snapshots. We then can predict the solution of $\boldsymbol{z}$ at time t^{N+1} as follows:

$$\boldsymbol{z}_{N+1} = \Phi_r e^{\Omega t^{N+1}} \boldsymbol{b}_r \tag{4.6}$$

where $\Omega := \frac{1}{\delta t} \ln(\Lambda)$ and $\boldsymbol{b}_r := \Phi_r^\dagger \boldsymbol{z}_1$.

5 Predictions of capsule deformation for query configuration

The construction of the ROM model from a set of configurations allows constructing a common basis of modes that correspond to an admissible set of solutions. This allows us to predict, not just the already known capsule deformation but also any admissible configuration using some manifold walking schemes (Paolella, 2018; Dikshit & Powar, 1982; Savignat, 2000; Nayroles *et al.*, 1992). The key idea of this approach is based on searching approximated solution using an admissible manifold of solutions. The prediction of query configuration, that is not stored in the database, consists of finding its corresponding principal components using some manifold-based prediction approaches such as Surface Interpolation approach (SI) and Diffuse Approximation approach (DA).

5.1 Surface Interpolation (SI) approach

The key idea behind applying this technique is to construct a hypersurface $S_m(\boldsymbol{\theta})$ for a set of principal components (modes) $m = 1, \ldots, r$ at time t and $\boldsymbol{\theta}$ is a parameter vector that will be defined in next chapter. From the surface S_m, a query mode ψ_q can be obtained for any desired configuration $\boldsymbol{\psi}_q = S_m(\boldsymbol{\theta}_q)$. To achieve this goal, a piecewise interpolation can be employed. In this work, we use linear interpolation and a cubic spline interpolation (Hastie *et al.*, 2009) which provides a smoother surface due to the continuity of the first derivatives and the cross-derivative of S_m.

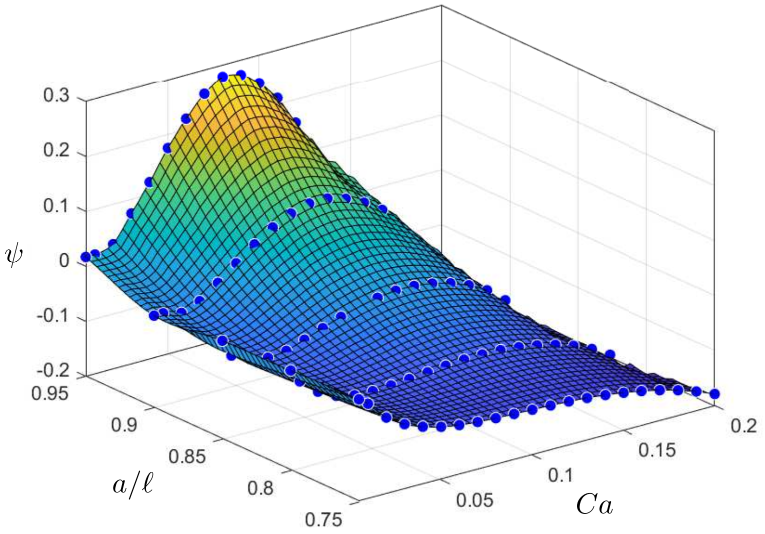


Figure 5.1: A manifold of admissible mode components ψ of all the capsule shapes. The blue dots are the database parameter modes.

As an example, figure 5.1 illustrates a manifold created from the values of $\boldsymbol{\psi}$ at Ca and a/l using the cubic interpolation approach. The construction of a manifold of admissible coefficients allows determining a configuration of interest $\boldsymbol{\psi}_q$ which will be located by the projection of the desired configuration $\boldsymbol{\theta}_q$ on the manifold. This procedure will be repeated for a fixed number of modes m. From this latter and with the common principal component vectors, the shape of the capsule for the desired configuration can be constructed by the formula 3.1.

5.2 Diffuse Approximation (DA) method

The diffuse approximation approach, also known as moving least squares (MLS) approximation, is a method of reconstructing continuous functions from a parametric space via the calculation of a weighted least squares measure biased towards the region around the point at which the reconstructed value is requested (Lancaster & Salkauskas, 1981; Nayroles *et al.*, 1992; Breitkopf *et al.*, 2005). The concept of diffuse approximation is about introducing a local weighted least squares fitting in a small neighborhood of a point and based on m-th nodes close to this point.

To local approximate a function f at a point $\boldsymbol{x}$, based on a set of sampling points $S = \{(x_i, f_i) \, | f(\boldsymbol{x}_i) = f_i\}$. The unknown f is approximated in the vicinity of $\boldsymbol{x}$ by

$$f(\boldsymbol{x}) \simeq f_{app} = p^T(\boldsymbol{x})\, \boldsymbol{a}(\boldsymbol{x}) \tag{5.1}$$

where $p^T(\boldsymbol{x})$ is a polynomial basis function and $\boldsymbol{a}$ the matrix of coefficients that are expressed by the minimization of the norm of the weighted difference between the estimated value at nodes and the nodal values f_i.

$$J_x(\boldsymbol{a}) = \frac{1}{2}\sum_i w_i(x_i, x)\left(\boldsymbol{p}^T(x_i)\,\boldsymbol{a} - f_i\right)^2 \tag{5.2}$$

where w is the weighting function that weights each nodal value to the approximation.

There are different ways to choose the weighting function. A simple choice for this latter is to choose a polynomial function rather than an exponential function as $exp(-d^2)$ for computation purposes. The weight functions are constructed from the reference windows functions w_{ref} such that it verifies $w_{ref}(x_i,\cdot) > 0$ inside the domain of interest of the node i and $w_{ref}(x_i,\cdot) = 0$ otherwise. A polynomial function

$$w_{ref} = \begin{cases} 2d^3 - 3d^2 + 1, & \text{if } d \le 1 \\ 0, & \text{otherwise} \end{cases},$$

Unlike surface interpolation and polynomial regression approaches, which operate one by one with the principal components of the manifold, DA (Nayroles *et al.*, 1992) is used as a method for accurately mapping a point $\boldsymbol{\theta}'$ in the parametric space $\mathcal{P}$ into a point in the manifold.

For relaibility reasons, in the construction of our model, we give preference to DA method over SI method.

6 Elements of analysis - Accuracy criteria

In order to measure the approximation error generated by the data dimensionality process, we introduce the classical Relative Information Content (RIC) (see for example (Silva & Alvaro, 2015)), which is computed as:

$$\mathrm{RIC}(K) = \frac{\sum\limits_{k=K+1}^{r} \tilde{\sigma}_k^2}{\sum\limits_{k=1}^{r} \tilde{\sigma}_k^2}, \quad K = 1,\ldots,r, \tag{6.1}$$

where $\tilde{\sigma}_k$ is the k-th singular value from the SVD decomposition, r is the rank of the matrix of study ($\mathcal{S}_\varrho$ or $\mathcal{T}_\varrho$) and K is the truncation rank. A supplementary indicator is the ratio

$$K \mapsto \frac{\tilde{\sigma}_K}{\tilde{\sigma}_1} \tag{6.2}$$

that gives an idea of the decay rate of the singular values.

The second criterion directly measures the error between the shape predicted by the ROM and the shape computed by the FOM. This is achieved by using the so-called Modified Hausdorff distance d_{MH} (Dubuisson & Jain, 1994) that we normalize by the capsule radius a. The modified Hausdorff distance computes the distance between two finite sets $\mathcal{F}$ and $\mathcal{G}$ of a normed space of norm $\|.\|$, and is defined as

$$d_{MH}\left(\mathcal{F},\mathcal{G}\right) = \max\left(d_h\left(\mathcal{F},\mathcal{G}\right), d_h\left(\mathcal{G},\mathcal{F}\right)\right), \tag{6.3}$$

with

$$d_h\left(\mathcal{F},\mathcal{G}\right) = \frac{1}{N_{\mathcal{F}}} \sum_{p_{\mathcal{F}} \in \mathcal{F}} d_s\left(p_{\mathcal{F}},\mathcal{G}\right) \tag{6.4}$$

where $N_{\mathcal{F}}$ is the number of points in the set $\mathcal{F}$ and $d_s\left(p_{\mathcal{F}},\mathcal{G}\right)$ is the distance between $p_{\mathcal{F}}$ and the set $\mathcal{G}$, which is defined as

$$d_s\left(p_{\mathcal{F}},\mathcal{G}\right) = \min_{p_{\mathcal{G}} \in \mathcal{G}} \|p_{\mathcal{F}} - p_{\mathcal{G}}\|. \tag{6.5}$$

7 Conclusion

In this chapter, we have presented the statement of the problem, the resolution strategy of the full order model, and the main tools of order reduction models. This latter will be used in the next chapter to introduce a new numerical method that can replace the FSI resolution.

Chapter 4

Deformation prediction of a capsule flowing into a microfluidic channel

Contents

1 Introduction

In this chapter, we propose two data-driven reduced-order modeling variants to simulate the capsule deformability behavior in a square channel. The first one consists of predicting capsule dynamics using a combination of ROMs and manifold learning. This combination is for a goal to deal with the space-time-parameter simulations database (Figure 1.1) that are pre-computed from FOM (section 2.2).

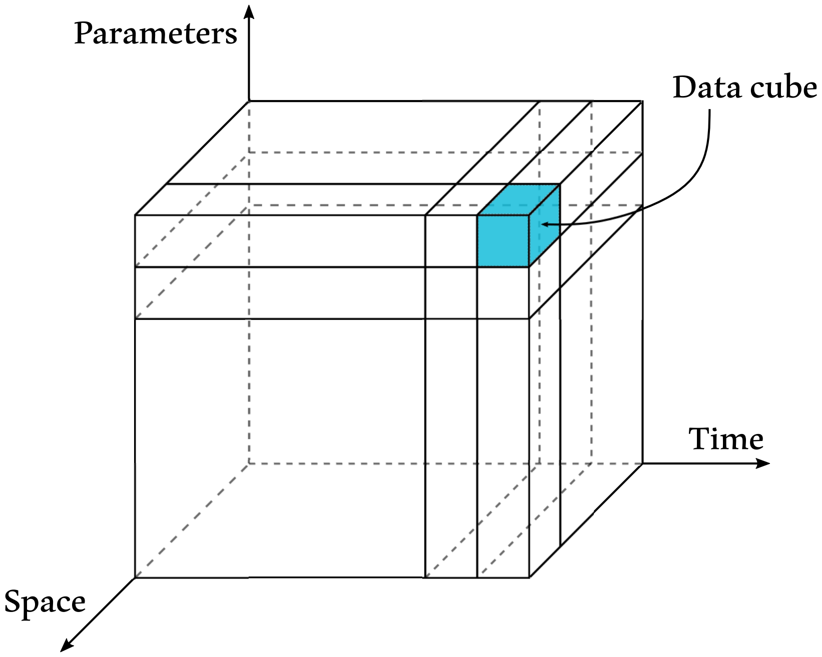


Figure 1.1: Space-time-parameter data cube

The second model is a dynamic data-driven reduced-order model. It describes the deformability evolution of a capsule in the form of a dynamic system using POD of the position and velocity pre-computed FOM simulations. Then, a DMD base model is used to determine the Koopman operator that allows predicting the capsule deformability at time t^{n+1} from its previous state at time t^n.

2 Database of FOM results

2.1 Design of experiment

Simulations of the FOM problem have been run varying the two governing parameters in the range $[0; 0.2]$ for the capillary numberCa and $[0.75; 1.2]$ for the confinement ratio a/ℓ. Only the simulations for which a steady-state shape was reached were retained in the database. The resulting numerical database, composed of $N_c = 118$ $(Ca, a/l)$ samples (Figure 2.1), describes the time-evolution of the three-dimensional position vectors of the capsule membrane nodes (Figure 2.2).

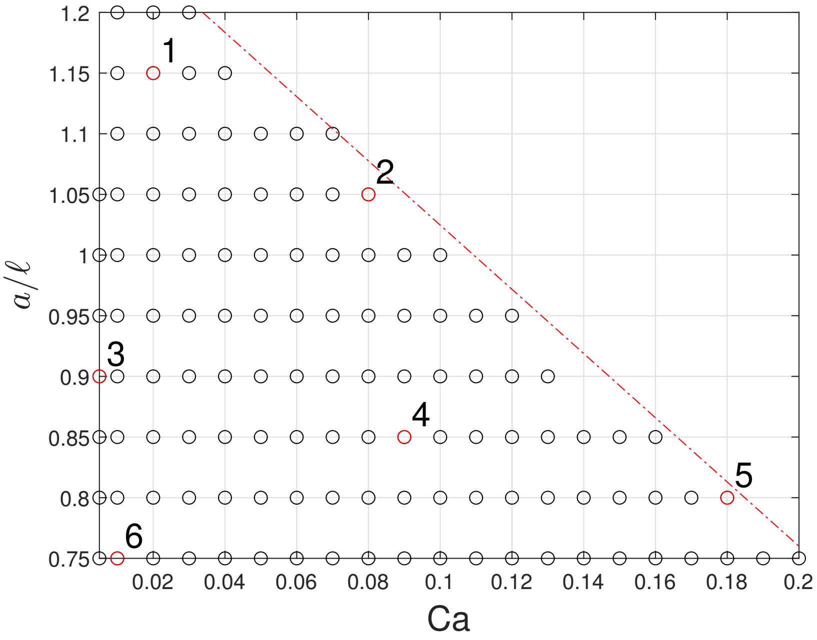


Figure 2.1: Values of Ca and a/l that are included in the FOM database for which an initially spherical capsule flows in a square-section microfluidic channel. No steady-state deformation can be obtained above the red dotted line for the neo-Hookean constitutive law.

Above the red dotted line in figure 2.1, the capsules exhibiting continuous elongation owing to the strain-softening behavior of the membrane law (Barthès-Biesel, 2011).

3 Data-driven model reduced-order model with manifold learning

The proposed model involves the construction of two PODs: parametric and spatial reduced-order models. A global reduced-order basis has then introduced the displacement and velocity of a capsule at a selection of snapshots (section 3). This allows introducing a reduced model that corresponds to any parameter vector of Ca and a/ℓ values by estimating its corresponding principal components (section 3.4). Then, with the use of a Diffuse Approximation (DA) method, we adopt a data-driven manifold learning to predict the deformation of the capsule in the flow for a chosen time discretization. A numerical validation demonstrates the new model's efficiency and accuracy compared to the full-order simulations. A sensibility analysis is also provided to prove the robustness of the approach.

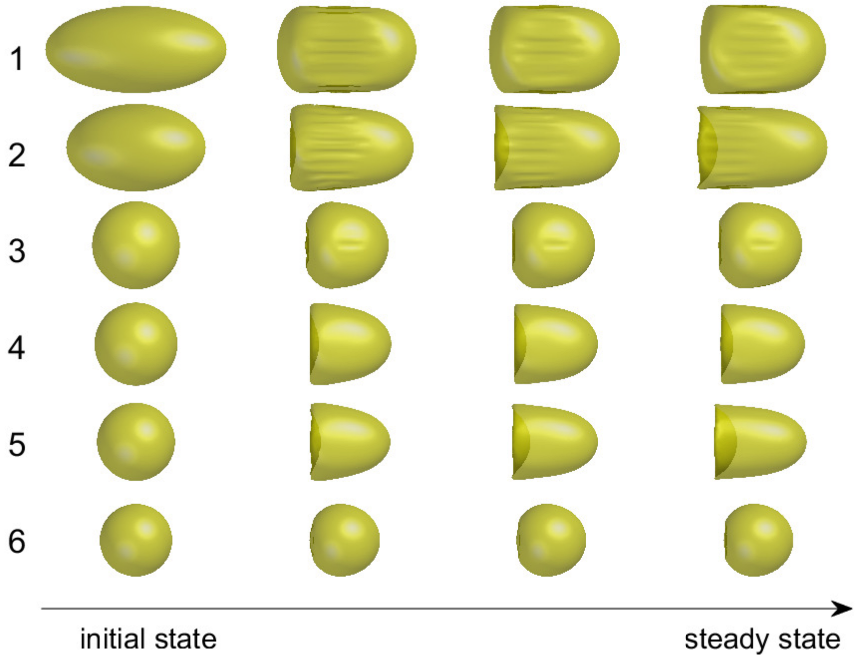


Figure 2.2: Illustration for time evolution cases of capsule deformations flowing in microfluidic channel with numbers in figure 2.1. The capsule is pre-deformed into an ellipsoid when $a/\ell \geq 0.95$.

3.1 Overview

We give below a short overview of the proposed data-driven model order reduction methodology. The approach is divided classically into two stages: An offline stage for the search of the principal components and POD coefficient matrices of the FOM solutions, then an online stage where a low-order system is used to simulate the problem for a configuration of parameter $(Ca, a/\ell)$.

1. **Offline stage**. We build two POD bases for space and parameter variables. The displacement field is represented as

$$\{\boldsymbol{u}\}(\{\boldsymbol{X}\}, t, \boldsymbol{\theta}) = \sum_{k=1}^{K_u^x} \sum_{\ell=1}^{K_u^c} A_{k\ell}(t) \, \{\boldsymbol{\Phi}_u^r\}_k \, (\boldsymbol{\psi}_u(\boldsymbol{\theta}))_\ell, \tag{3.1}$$

   where $\{\boldsymbol{\Phi}_u^r\}_k \in \mathbb{R}^{3N_x}$ are the spatial POD modes, $\boldsymbol{\psi}_u(\boldsymbol{\theta}) \in \mathbb{R}^{K_u^c}$ the parameter modes and $A_{k\ell}(t)$ scalar coefficients depending on time t. The truncation ranks are K_u^x and K_u^c, respectively (the $'x'$ superscript stands for 'space' and the $'c'$ superscript for 'configuration'). We use a similar representation for the velocity field:

$$\{\boldsymbol{v}\}(\{\boldsymbol{X}\}, t, \boldsymbol{\theta}) = \sum_{k=1}^{K_v^x} \sum_{\ell=1}^{K_v^c} B_{k\ell}(t) \, \{\boldsymbol{\Phi}_v^r\}_k \, (\boldsymbol{\psi}_v(\boldsymbol{\theta}))_\ell \, . \tag{3.2}$$

The two POD bases are determined using singular value decomposition (SVD) from the datacube with different data rearrangements in stacked matrix form. The truncation ranks K_u^x, K_u^c, K_v^x, K_v^c are expected to be rather small while ensuring the accuracy of the representations.

2. **Online stage**. For any query configuration of parameters $\boldsymbol{\theta}_q$ in the parameter domain:

   (a) Estimate the displacement field $\{\boldsymbol{u}\}(\{\boldsymbol{X}\},t,\boldsymbol{\theta_q})$ from expression (3.1) through an interpolation process at $\boldsymbol{\theta} = \boldsymbol{\theta}_q$. For that, we use a diffuse approximation technique (Breitkopf *et al.*, 2002) that can be used for any parameter space dimension;

   (b) From the displacement $\{\boldsymbol{u}\}(\{\boldsymbol{X}\},t^i,\boldsymbol{\theta_q})$ and velocity $\{\boldsymbol{v}\}(\{\boldsymbol{X}\},t^i,\boldsymbol{\theta_q})$ fields estimated at different instants $t^i \in [0,T_f]$, compute their corresponding low-order reduced basis by singular value decomposition. Then, we get the low-order representations of both displacements and velocities:

$$\{\boldsymbol{u}\}(\{\boldsymbol{X}\},t,\boldsymbol{\theta_q}) = \sum_{k=1}^{m_u} \alpha_k(t)\ \{\boldsymbol{\varphi}^k\}(\boldsymbol{\theta}_q), \tag{3.3}$$

$$\{\boldsymbol{v}\}(\{\boldsymbol{X}\},t,\boldsymbol{\theta_q}) = \sum_{k=1}^{m_v} \xi_k(t)\ \{\gamma^k\}(\boldsymbol{\theta}_q), \tag{3.4}$$

   (c) Manifold learning online stage: with the use of diffuse approximation method, we determine the low-order manifold $\mathcal{M}$ that relies displacements and velocities in the (reduced-order) state space:

$$\boldsymbol{\xi} = \mathcal{M}(\boldsymbol{\alpha},\, \boldsymbol{\theta}_q);$$

   (d) Derivation of a low-order dynamical system: we then derive a lightweight differential-algebraic dynamical system, for $\boldsymbol{\theta} = \boldsymbol{\theta}_q$, solve

$$\begin{aligned} \frac{d\boldsymbol{\alpha}}{dt} &= Q\,\boldsymbol{\xi}(t), \\ \boldsymbol{\xi}(t) &= \mathcal{M}(\boldsymbol{\alpha}(t),\, \boldsymbol{\theta}_q). \end{aligned}$$

   The high-dimensional displacement and velocity fields can then be reconstructed according to (3.3) and (3.4).

In the next section, we give all the details of the ROM methodology.

3.2 Offline stage

3.2.1 Global Parametric Reduced Basis (GPRB)

This first step consists in computing a parametric reduced basis in the whole parameter domain from the database of FOM results (see Section 2.3). For simplification reasons, we use the subscript ϱ that can be either u or v to express displacements and velocity respectively in the formulas.

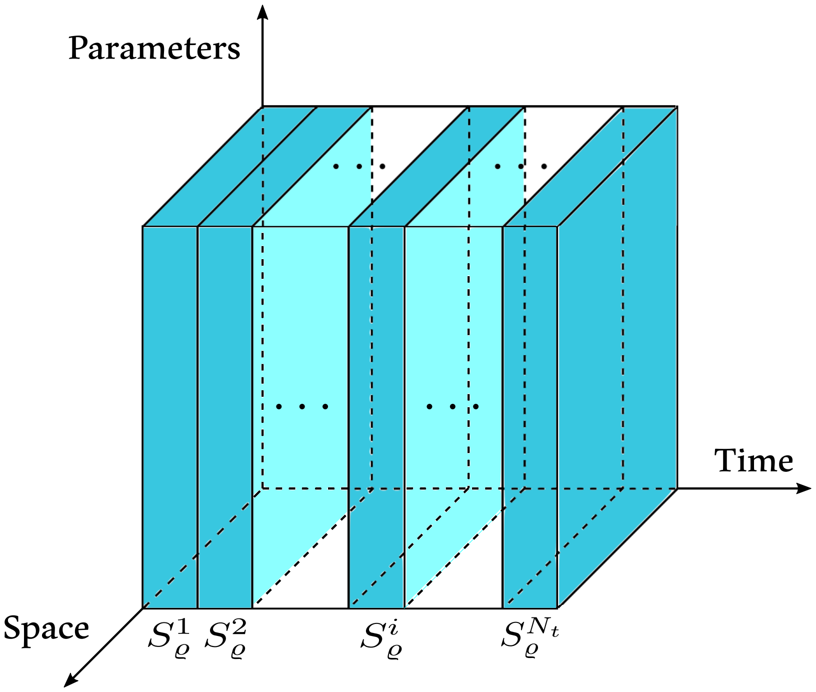


Figure 3.1: FOM data rearrangements for parametric data set selection.

Let $\boldsymbol{S}_u^i \in \mathcal{M}_{3N_x,N_c}(\mathbb{R})$ be the matrix of capsule displacement fields $\{\boldsymbol{u}\}$ and $\boldsymbol{S}_v^i \in \mathcal{M}_{3N_x,N_c}(\mathbb{R})$ the matrix of the velocity fields $\{\boldsymbol{v}\}$ at time t^i, $i = 1,...,N_t$ (Figure 3.2a), considering all the configurations $\boldsymbol{\theta}_j$ for $j = 1,\dots,N_c$ of the database, i.e.

$$\boldsymbol{S}_u^i = \left[\{\boldsymbol{u}\}\left(\{\boldsymbol{X}\},t^i,\boldsymbol{\theta}_1\right),\ \dots,\ \{\boldsymbol{u}\}\left(\{\boldsymbol{X}\},t^i,\boldsymbol{\theta}_{N_c}\right)\right],$$

and

$$\boldsymbol{S}_v^i = \left[\{\boldsymbol{v}\}\left(\{\boldsymbol{X}\},t^i,\boldsymbol{\theta}_1\right),\ \dots,\ \{\boldsymbol{v}\}\left(\{\boldsymbol{X}\},t^i,\boldsymbol{\theta}_{N_c}\right)\right].$$

Then we stack all the matrices $\boldsymbol{S}_\varrho^i$ for $i = 1,\dots,N_t$ into a big matrix $\mathcal{S}_\varrho \in \mathcal{M}_{3N_x\times N_t,N_c}(\mathbb{R})$:

$$\mathcal{S}_\varrho = \begin{bmatrix} \boldsymbol{S}_\varrho^1 \\ \boldsymbol{S}_\varrho^2 \\ \vdots \\ \boldsymbol{S}_\varrho^{N_t} \end{bmatrix} \text{ for } \varrho = u, v.$$

We then apply SVD (Golub & Reinsch, 1971) and get:

$$\mathcal{S}_\varrho = U_\varrho \Sigma_{S_\varrho} \Psi_\varrho^T, \text{ for } \varrho = u, v, \tag{3.5}$$

where $U_\varrho \in \mathcal{M}_{3N_xN_t,N_c}(\mathbb{R})$, $\Psi_\varrho \in \mathcal{M}_{N_c}(\mathbb{R})$ are semi-orthogonal and orthogonal matrices, respectively, and $\Sigma_{S_\varrho} \in \mathcal{M}_{N_c}(\mathbb{R})$ is the diagonal singular value matrix. The matrix Ψ_ϱ of discrete parameter modes can be truncated according to K_ϱ^c parameters, so we note:

$$\begin{cases} \Psi_\varrho^r = \left[(\Psi_\varrho)_1, ..., (\Psi_\varrho)_{K_\varrho^c} \right] \in \mathcal{M}_{N_c, K_\varrho^c}(\mathbb{R}), \\ \text{with } (\Psi_\varrho)_k \in \mathcal{M}_{N_c,1}(\mathbb{R}) \text{ for } k = 1, \dots, K_\varrho^c, \text{ and } \varrho = u, v. \end{cases} \tag{3.6}$$

The orthogonality property ensures that $\left(\Psi_\varrho^r\right)^T \Psi_\varrho^r = I_{K_\varrho^c}$.

3.2.2 Global Spatial Reduced Basis (GSRB)

Similarly, we build a global spatial reduced basis that captures the spatial data of capsule displacements.

Let $\boldsymbol{T}_u^j \in \mathcal{M}_{3N_x,N_t}(\mathbb{R})$ be the displacement matrix and $\boldsymbol{T}_v^j$ the velocity matrix for the j-th configuration $\boldsymbol{\theta}_j$, for $j = 1, \dots, N_c$ at all time instants t^i, $i = 1, \dots, N_t$ (Figure 3.2b):

$$\boldsymbol{T}_u^j = \left[\{\boldsymbol{u}\}\left(\{\boldsymbol{X}\}, t^1, \boldsymbol{\theta}_j\right), \dots, \{\boldsymbol{u}\}\left(\{\boldsymbol{X}\}, t^{N_t}, \boldsymbol{\theta}_j\right) \right],$$

and

$$\boldsymbol{T}_v^j = \left[\{\boldsymbol{v}\}\left(\{\boldsymbol{X}\}, t^1, \boldsymbol{\theta}_j\right), \dots, \{\boldsymbol{v}\}\left(\{\boldsymbol{X}\}, t^{N_t}, \boldsymbol{\theta}_j\right) \right].$$

Then we define the global matrix $\mathcal{T}_\varrho \in \mathcal{M}_{3N_x,N_t \times N_c}(\mathbb{R})$ that horizontally gathers all the matrices $\boldsymbol{T}_\varrho^j$ for $j = 1, ..., N_c$ and $\varrho = u, v$, respectively:

$$\mathcal{T}_\varrho = \left[\boldsymbol{T}_\varrho^1, \boldsymbol{T}_\varrho^2, \dots, \boldsymbol{T}_\varrho^{N_c} \right], \text{ for } \varrho = u, v.$$

The SVD decomposition is applied on $\mathcal{T}_\varrho$ to get

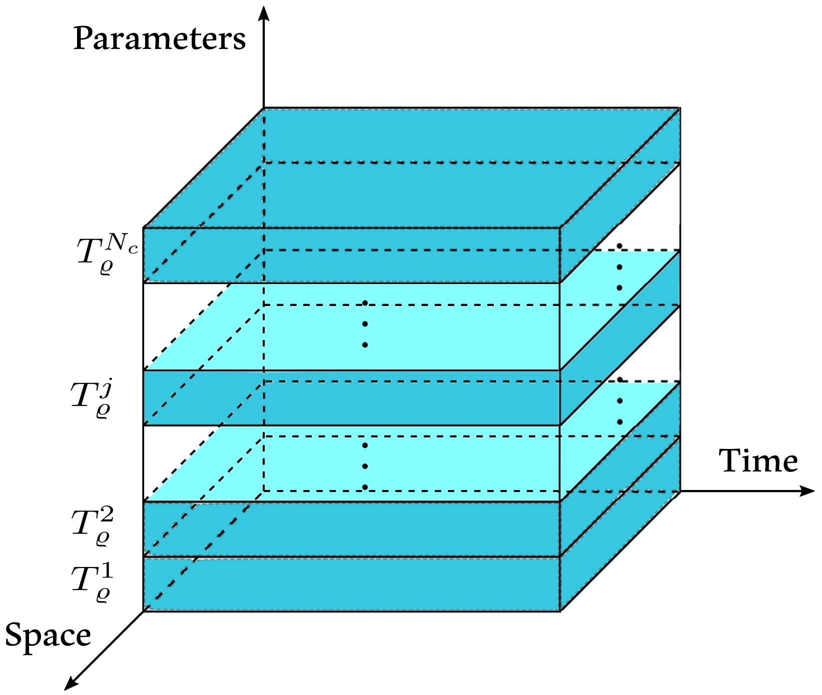


Figure 3.2: FOM data rearrangements for spatial data set selection.

$$\mathcal{T}_\varrho = \Phi_\varrho \, \Sigma_{T_\varrho} \, V_\varrho^T, \text{ for } \varrho = u, v, \tag{3.7}$$

where $\Phi_\varrho \in \mathcal{M}_{3N_x}(\mathbb{R})$, $V_\varrho \in \mathcal{M}_{N_cN_t,3N_x}(\mathbb{R})$ are orthogonal and semi-orthogonal matrices, respectively, and $\Sigma_{S_\varrho} \in \mathcal{M}_{3N_x}(\mathbb{R})$ is the diagonal singular value matrix with singular values organized in decreasing order. We can also apply a spatial basis truncation at a range of K_ϱ^x for a specified accuracy threshold. The reduced spatial POD basis is stored in the matrix:

$$\Phi_\varrho^r = \left[\{\boldsymbol{\phi}_\varrho\}_1, \ldots, \{\boldsymbol{\phi}_\varrho\}_{K_\varrho^x} \right] \in \mathcal{M}_{3N_x,K_\varrho^x}(\mathbb{R}) \tag{3.8}$$

with the orthogonality property $\left(\Phi_\varrho^r\right)^T \Phi_\varrho^r = I_{K_\varrho^x}$, $\varrho = u, v$.

3.3 Data dimensionality reduction

Once the POD modes of $\mathcal{S}_\varrho$ and $\mathcal{T}_\varrho$ for the displacement fields ($\varrho = u$) and the velocity fields ($\varrho = v$) are computed, one can summarize (approximate) capsule displacement and velocity fields of the database at any discrete time t^i ($i = 1 \ldots, N_t$) as

$$\{\boldsymbol{u}\} \left(\{\boldsymbol{X}\},\, t^i,\, [\boldsymbol{\theta}_1, ..., \boldsymbol{\theta}_{N_c}] \right) \approx \Phi_u^r \, A(t^i) \, (\Psi_u^r)^T \in \mathcal{M}_{3N_x,N_c}(\mathbb{R}), \tag{3.9}$$

$$\{\boldsymbol{v}\}\left(\{\boldsymbol{X}\}, t^i, [\boldsymbol{\theta}_1, ..., \boldsymbol{\theta}_{N_c}]\right) \approx \Phi_v^r \, B(t^i) \, (\Psi_v^r)^T \in \mathcal{M}_{3N_x, N_c}(\mathbb{R}), \tag{3.10}$$

where $A(t^i) \in \mathcal{M}_{K_u^x, K_u^c}(\mathbb{R})$ and $B\left(t^i\right) \in \mathcal{M}_{K_v^x, K_v^c}(\mathbb{R})$ are some coefficient matrices depending on time t^i. If the approximation is chosen as the orthogonal projection over the vector spaces spanned by the POD modes, the coefficient matrices are computed as follows for $i = 1 \ldots, N_t$:

$$A(t^i) = \underbrace{(\Phi_u^r)^T}_{K_u^x \times (3N_x)} \; \underbrace{\{\boldsymbol{u}\}\left(\{\boldsymbol{X}\}, t^i, [\boldsymbol{\theta}_1, ..., \boldsymbol{\theta}_{N_c}]\right)}_{(3N_x) \times N_c} \; \underbrace{\Psi_u^r}_{N_c \times K_u^c}, \tag{3.11}$$

$$B(t^i) = \underbrace{(\Phi_v^r)^T}_{K_v^x \times (3N_x)} \; \underbrace{\{\boldsymbol{v}\}\left(\{\boldsymbol{X}\}, t^i, [\boldsymbol{\theta}_1, ..., \boldsymbol{\theta}_{N_c}]\right)}_{(3N_x) \times N_c} \; \underbrace{\Psi_v^r}_{N_c \times K_v^c}. \tag{3.12}$$

The outputs of the offline stage are respectively the POD matrices Φ_u^r, Φ_v^r, Ψ_u^r, Ψ_v^r and the small matrices $A(t^i)$, $B(t^i)$, $i = 1, ..., N_t$. The next online stage will operate on the summarized data (3.9),(3.10) with coefficients matrices (3.11),(3.12). The algorithm of the offline phase is summarized in Algorithm 1.

Algorithm 1 Offline phase

```
Require: database of $\boldsymbol{\theta}_k$ for $k = 1, \ldots, N_c$, truncations $K^c_\varrho$, number of snapshots $N_t$.
  for $i \leftarrow 1, \ldots, N_t$ do
    if $(\varrho = u)$ then
      $\boldsymbol{S}^i_u \leftarrow \Big[\{\boldsymbol{u}\}\left(\{\boldsymbol{X}\}, t^i, \boldsymbol{\theta}_1\right), \ \ldots, \ \{\boldsymbol{u}\}\left(\{\boldsymbol{X}\}, t^i, \boldsymbol{\theta}_{N_c}\right)\Big]$; $\mathcal{S}_u \leftarrow [\mathcal{S}_u; \boldsymbol{S}^i_u]$;
    else
      $\boldsymbol{S}^i_v \leftarrow \Big[\{\boldsymbol{v}\}\left(\{\boldsymbol{X}\}, t^i, \boldsymbol{\theta}_1\right), \ \ldots, \ \{\boldsymbol{v}\}\left(\{\boldsymbol{X}\}, t^i, \boldsymbol{\theta}_{N_c}\right)\Big]$; $\mathcal{S}_v \leftarrow [\mathcal{S}_v, \boldsymbol{S}^i_v]$;
    end if
  end for
  for $j \leftarrow 1, \ldots, N_c$ do
    if $(\varrho = u)$ then
      $\boldsymbol{T}^j_u \leftarrow \Big[\{\boldsymbol{u}\}\left(\{\boldsymbol{X}\}, t^1, \boldsymbol{\theta}_j\right), \ \ldots, \ \{\boldsymbol{u}\}\left(\{\boldsymbol{X}\}, t^{N_t}, \boldsymbol{\theta}_j\right)\Big]$; $\mathcal{T}_u \leftarrow [\mathcal{T}_u, \boldsymbol{T}^j_u]$;
    else
      $\boldsymbol{T}^j_v \leftarrow \Big[\{\boldsymbol{v}\}\left(\{\boldsymbol{X}\}, t^1, \boldsymbol{\theta}_j\right), \ \ldots, \ \{\boldsymbol{v}\}\left(\{\boldsymbol{X}\}, t^{N_t}, \boldsymbol{\theta}_j\right)\Big]$; $\mathcal{T}_v \leftarrow [\mathcal{T}_v, \boldsymbol{T}^j_v]$;
    end if
  end for
  $\Phi_\varrho \leftarrow \mathrm{SVD}(\mathcal{S}_\varrho)$, $\Psi_\varrho \leftarrow \mathrm{SVD}(\mathcal{T}_\varrho)$, for $\varrho \leftarrow u, v$;
  for $i = 1, \ldots, N_t$ do
    $A(t^i) \leftarrow (\Phi^r_u)^T \ \{\boldsymbol{u}\}\left(\{\boldsymbol{X}\}, t^i, [\boldsymbol{\theta}_1, ..., \boldsymbol{\theta}_{N_c}]\right) \ \Psi^r_u$;
    $B(t^i) \leftarrow (\Phi^r_v)^T \ \{\boldsymbol{v}\}\left(\{\boldsymbol{X}\}, t^i, [\boldsymbol{\theta}_1, ..., \boldsymbol{\theta}_{N_c}]\right) \ \Psi^r_v$;
  end for
```

3.4 Online stage: search for an approximate solution

In the online stage, a user will ask for an approximate solution at a new (query) configuration $\boldsymbol{\theta} = \boldsymbol{\theta}_q$ that has not been already computed by the FOM solver or is not stored in the database. Ingredients of the online stage will be: i) the data summarization of the previous offline stage; ii) a first estimation of the spatio-temporal solution at $\boldsymbol{\theta} = \boldsymbol{\theta}_q$; iii) the computation of a low-dimensional spatial reduced basis suitable for $\boldsymbol{\theta} = \boldsymbol{\theta}_q$; iv) the construction of a manifold $\mathcal{M}$ that links variables of displacements and velocities in the low-order state space to solve the equation of membrane mechanics; v) finally, the building of a low-order differential-algebraic (DAE) system of equations that defines the reduced-order model. Substeps ii) and iv) will make use of diffuse approximation (DA) as a universal approximator for multivariate functions.

3.4.1 First estimation of the solutions at $\boldsymbol{\theta} = \boldsymbol{\theta}_q$

As an introduction, let us assume that, from the parameter sampling $\{\boldsymbol{\theta}_1, ..., \boldsymbol{\theta}_{N_c}\}$, we consider a polynomial Lagrange interpolation with Lagrange polynomials denoted by $L_i(\boldsymbol{\theta})$ such that the Lagrange property

$$L_i(\theta_j) = \delta_{ij}, \quad 1 \leq i, j \leq N_c$$

is fulfilled (δ_{ij} is the standard Kronecker symbol). Let us denote by $\boldsymbol{L}(\boldsymbol{\theta}) = (L_j(\boldsymbol{\theta}))_{j=1,...,N_c} \in \mathbb{R}^{N_c}$ the vector that stores the Lagrange polynomials. Then

$$\mathcal{I}\{\boldsymbol{u}\}\left(\{\boldsymbol{X}\},\, t^i,\, \boldsymbol{\theta}_q\right) := \{\boldsymbol{u}\}\left(\{\boldsymbol{X}\},\, t^i,\, [\boldsymbol{\theta}_1, ..., \boldsymbol{\theta}_{N_c}]\right) \boldsymbol{L}(\boldsymbol{\theta}_q) \in \mathbb{R}^{3N_x}$$

is an interpolated displacement field at parameter $\boldsymbol{\theta} = \boldsymbol{\theta}_q$ and discrete time $t = t^i$. One can of course do the same for the velocity field.

Unfortunately, Lagrange polynomial interpolation is not suitable for parameter spaces of arbitrary dimension because of the curse of dimensionality and because it may suffer from instability issues (Runge phenomenon). Rather than using polynomial interpolation, we propose to use a Diffuse Approximation (DA) technique (Breitkopf *et al.*, 2002; Quesada *et al.*, 2021) which is an approximation method based on local low-order polynomial reconstruction (of order one or two) using a compactly-supported kernel function and weighted least squares. The DA method is known to be a robust and reliable approach which is less sensitive to the location of the sampling points. Moreover, it can be applied to multivariate functions of arbitrary dimensions, which is interesting for larger or more general parameter spaces.

To estimate the displacement field for $\boldsymbol{\theta} = \boldsymbol{\theta}_q$, we look for a vector $\boldsymbol{\psi}_u(\boldsymbol{\theta}_q) \in \mathbb{R}^{K_u^c}$ such that

$$\{\boldsymbol{u}\}(\{\boldsymbol{x}\}, t^i, \boldsymbol{\theta}_q) = \Phi_u^r\, A(t^i)\, \boldsymbol{\psi}_u(\boldsymbol{\theta}_q) \tag{3.13}$$

returns an approximation of the displacement field at $\boldsymbol{\theta} = \boldsymbol{\theta}_q$. Similarly for the velocity field, we search for a vector $\boldsymbol{\psi}_v(\theta_q) \in \mathbb{R}^{K_v^c}$ that gives

$$\{\boldsymbol{v}\}(\{\boldsymbol{x}\}, t^i, \boldsymbol{\theta}_q) = \Phi_v^r\, B(t^i)\, \boldsymbol{\psi}_v(\boldsymbol{\theta}_q). \tag{3.14}$$

Each vector $\boldsymbol{\psi}_\varrho(\boldsymbol{\theta}_q) \in \mathbb{R}^{K_\varrho^c}$ can be locally approximated by

$$\Psi_\varrho(\boldsymbol{\theta}_q) = \mathcal{A}_\varrho\, \boldsymbol{p}(\boldsymbol{\theta}_q), \text{ for } \varrho = u, v, \tag{3.15}$$

where the matrix $\mathcal{A}_\varrho \in \mathcal{M}_{K_\varrho^c, m}(\mathbb{R})$ (to be determined) is the approximation coefficient matrix and $\boldsymbol{p}(\boldsymbol{\theta}_q) \in \mathbb{R}^m$ is a vector of independent polynomial functions, where

$$\begin{cases} \boldsymbol{p}(\boldsymbol{\theta}) = \begin{bmatrix} 1 & Ca & a/\ell \end{bmatrix}^T, \; m = 3 & \text{for first order DA,} \\ \boldsymbol{p}(\boldsymbol{\theta}) = \begin{bmatrix} 1 & Ca & a/\ell & Ca\,(a/\ell) & (Ca)^2 & (a/\ell)^2 \end{bmatrix}^T, \; m = 6 & \text{for second order DA.} \end{cases} \tag{3.16}$$

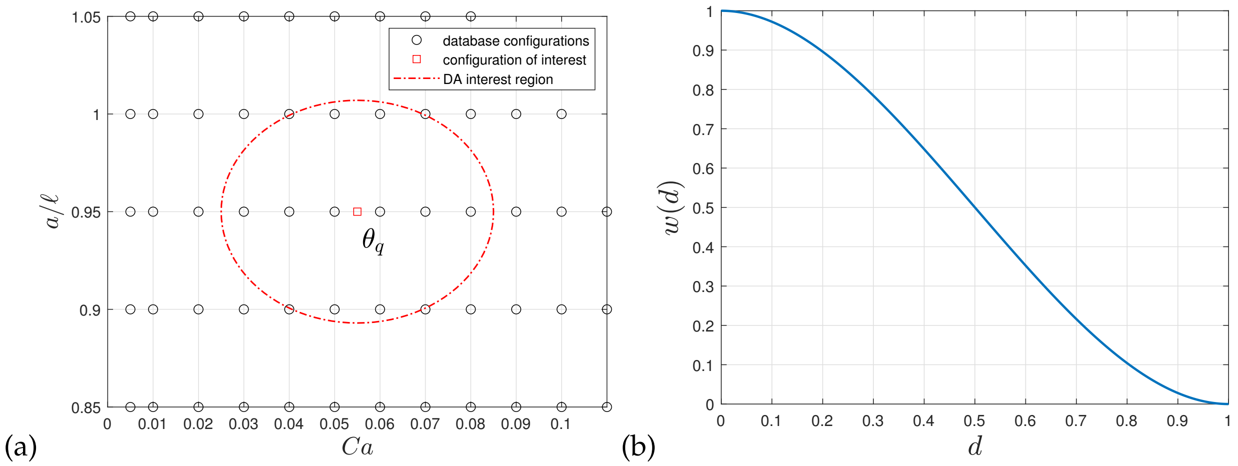


Figure 3.3: (a) DA elliptical region of interest (dashed line) defined around the point $\boldsymbol{\theta}_q = (Ca = 0.055, a/\ell = 0.95)$ in the parametric space with $M = 10$ neighbors; (b) Weight function $w(d)$.

To approximate $\boldsymbol{\psi}_\varrho(\boldsymbol{\theta}_q)$, let us consider a neighborhood $\mathscr{S}(\boldsymbol{\theta}_q)$ centered on $\boldsymbol{\theta}_q$ containing M neighboring points (Figure 3.3a). It is an ellipse of equation

$$\left(\theta_1 - (\boldsymbol{\theta}_q)_1\right)^2 + \tilde{r}^2 \left(\theta_2 - (\boldsymbol{\theta}_q)_2\right)^2 = R^2$$

where $\tilde{r}$ is fixed (equal to 1.9 in Figure 3.3a) and R is chosen such that the ellipse contains M points (M being chosen by the operator). In other words, the distance between $\boldsymbol{\theta} = (\theta_1, \theta_2)^T$ and $\boldsymbol{\theta}_q$ is

$$d = \left(\left(\theta_1 - (\boldsymbol{\theta}_q)_1\right)^2 + \tilde{r}^2 \left(\theta_2 - (\boldsymbol{\theta}_q)_2\right)^2\right)^{\frac{1}{2}} / R. \tag{3.17}$$

One can use the candidate weight function (Figure 3.3b)

$$\begin{cases} w\left(d\right) = 2\,d^3 - 3\,d^2 + 1, & d \leq 1, \\ 0, & \text{otherwise.} \end{cases} \tag{3.18}$$

Diffuse approximation consists in minimizing the weighted least square problem

$$\min_{\mathcal{A}_\varrho \in \mathcal{M}_{K^c_\varrho, m}(\mathbb{R})} \quad J_{\boldsymbol{\theta}_q}\left(\mathcal{A}_\varrho\right) := \frac{1}{2} \sum_{\boldsymbol{\theta} \in \mathscr{S}(\boldsymbol{\theta}_q)} w\left(d(\boldsymbol{\theta})\right) \left\| \mathcal{A}_\varrho\, \boldsymbol{p}\left(\boldsymbol{\theta}\right) - \left[\Psi^r_\varrho\left(\boldsymbol{\theta}\right)\right]^T \right\|^2_{\mathbb{R}^{K^c_\varrho}} \tag{3.19}$$

where $[\Psi^r_\varrho\left(\boldsymbol{\theta}\right)]^T$ is the truncated matrix of modes that correspond to couples $\boldsymbol{\theta}_k$, $k = 1, \ldots, N_c$. The solution $\mathcal{A}_\varrho$ ($\varrho = u, v$) of the weighted least square problem (3.19) is then

$$\mathcal{A}_\varrho = (\Psi^r_\varrho)^T \mathcal{W}\mathcal{P} \left(\mathcal{P}^T \mathcal{W}\mathcal{P}\right)^{-1} \in \mathcal{M}_{K^c_\varrho, m}(\mathbb{R}) \tag{3.20}$$

where the matrix $\mathcal{P} \in \mathcal{M}_{N_c, m}\left(\mathbb{R}\right)$ and the diagonal matrix of weights $\mathcal{W} \in \mathcal{M}_{N_c}\left(\mathbb{R}\right)$ are defined as

$$\mathcal{P} = \begin{bmatrix} \boldsymbol{p}\left(\boldsymbol{\theta}_1\right)^T \\ \vdots \\ \boldsymbol{p}\left(\boldsymbol{\theta}_{N_c}\right)^T \end{bmatrix} \text{ and } \mathcal{W} = \begin{bmatrix} w_1 & 0 & \cdots & 0 \\ 0 & w_2 & & \vdots \\ \vdots & & \ddots & \\ 0 & \cdots & & w_{N_c} \end{bmatrix}. \tag{3.21}$$

3.4.2 Construction of a low-order reduced basis suitable for $\boldsymbol{\theta} = \boldsymbol{\theta}_q$, data generation

From (3.13) and (3.14), one can easily generate some pseudo-snapshot matrices $\mathcal{U}(\boldsymbol{\theta}_q)$ and $\mathcal{V}(\boldsymbol{\theta}_q)$ that gather the estimated displacements and velocities at N_t discrete times, respectively:

$$\begin{cases} \mathcal{U}(\boldsymbol{\theta}_q) = \left[\{\boldsymbol{u}\}\left(\{\boldsymbol{X}\}, t^1, \boldsymbol{\theta}_q\right), \ldots, \{\boldsymbol{u}\}\left(\{\boldsymbol{X}\}, t^{N_t}, \boldsymbol{\theta}_q\right) \right], \\ \mathcal{V}(\boldsymbol{\theta}_q) = \left[\{\boldsymbol{v}\}\left(\{\boldsymbol{X}\}, t^1, \boldsymbol{\theta}_q\right), \ldots, \{\boldsymbol{v}\}\left(\{\boldsymbol{X}\}, t^{N_t}, \boldsymbol{\theta}_q\right) \right]. \end{cases} \tag{3.22}$$

One can then apply a new SVD decomposition of matrices $\mathcal{U}(\boldsymbol{\theta}_q)$ and $\mathcal{V}(\boldsymbol{\theta}_q)$ respectively to get spatial POD modes $\{\boldsymbol{\phi}^k\}(\boldsymbol{\theta}_q) \in \mathbb{R}^{3N_x}$, $k = 1, \ldots, m_u$ for $\{\boldsymbol{u}\}$ and velocity POD modes $\{\gamma^k\}(\boldsymbol{\theta}_q) \in \mathbb{R}^{3N_x}$, $k = 1, \ldots, m_v$ for $\{\boldsymbol{v}\}$.

$$\text{POD}\left(\mathcal{U}(\boldsymbol{\theta}_q)\right) \quad \rightarrow \quad \{\boldsymbol{\phi}^1\}(\boldsymbol{\theta}_q), \ldots, \{\boldsymbol{\phi}^{m_u}\}(\boldsymbol{\theta}_q) \tag{3.23}$$

$$\text{POD}\left(\mathcal{V}(\boldsymbol{\theta}_q)\right) \quad \rightarrow \quad \{\gamma^1\}(\boldsymbol{\theta}_q), \ldots, \{\gamma^{m_v}\}(\boldsymbol{\theta}_q) \tag{3.24}$$

where m_u and m_v are the truncation ranks of displacement and velocity modes determined in the next section on numerical experiments. One can then search the displacement and velocity fields at $\boldsymbol{\theta} = \boldsymbol{\theta}_q$ as

$$\{\boldsymbol{u}\}\left(\{\boldsymbol{X}\}, t, \boldsymbol{\theta}_q\right) = \sum_{k=1}^{m_u} \alpha_k\left(t\right)\{\boldsymbol{\varphi}^k\}\left(\boldsymbol{\theta}_q\right), \tag{3.25}$$

$$\{\boldsymbol{v}\}\left(\{\boldsymbol{X}\}, t, \boldsymbol{\theta}_q\right) = \sum_{k=1}^{m_v} \xi_k\left(t\right)\{\gamma^k\}\left(\boldsymbol{\theta}_q\right). \tag{3.26}$$

By denoting

$$\Phi(\boldsymbol{\theta}_q) = \left[\{\boldsymbol{\varphi}^1\}\left(\boldsymbol{\theta}_q\right), ..., \{\boldsymbol{\varphi}^{m_u}\}\left(\boldsymbol{\theta}_q\right)\right] \in \mathcal{M}_{3Nx,m_u}(\mathbb{R}), \tag{3.27}$$

$$\Gamma(\boldsymbol{\theta}_q) = \left[\{\gamma^1\}\left(\boldsymbol{\theta}_q\right), ..., \{\gamma^{m_v}\}\left(\boldsymbol{\theta}_q\right)\right] \in \mathcal{M}_{3Nx,m_v}(\mathbb{R}) \tag{3.28}$$

and $\boldsymbol{\alpha}(t) = [\alpha_1(t), ..., \alpha_{m_u}(t)]^T \in \mathbb{R}^{m_u}$, $\boldsymbol{\xi}(t) = [\xi_1(t), ..., \xi_{m_v}(t)]^T \in \mathbb{R}^{m_v}$, we have the vector formulas

$$\{\boldsymbol{u}\}\left(\{\boldsymbol{X}\}, t, \boldsymbol{\theta}_q\right) = \Phi(\boldsymbol{\theta}_q)\,\boldsymbol{\alpha}(t), \quad \{\boldsymbol{v}\}\left(\{\boldsymbol{X}\}, t, \boldsymbol{\theta}_q\right) = \Gamma(\boldsymbol{\theta}_q)\,\boldsymbol{\xi}(t). \tag{3.29}$$

The mode matrices $\Phi(\boldsymbol{\theta}_q)$ and $\Gamma(\boldsymbol{\theta}_q)$ are assumed to be orthonormal (w.r.t the natural Euclidean inner product), so we have $[\Phi(\boldsymbol{\theta}_q)]^T\,\Phi(\boldsymbol{\theta}_q) = I_{m_u}$ and $[\Gamma(\boldsymbol{\theta}_q)]^T\,\Gamma(\boldsymbol{\theta}_q) = I_{m_v}$.

3.4.3 Toward a physically consistent dynamical reduced-order model

Consider now the forward Euler scheme on the FSI system with a ROM time step $\delta t^{ROM} > 0$: at time $t^{i+1,ROM} = t^{i,ROM} + \delta t^{ROM}$, the numerical scheme is

$$\{\boldsymbol{u}^{i+1}\} = \{\boldsymbol{u}^i\} + \delta t^{ROM}\,\{\boldsymbol{v}^i\}, \tag{3.30}$$

$$\{\boldsymbol{v}^{i+1}\} = \{\boldsymbol{\varphi}\}(\{\boldsymbol{u}^{i+1}\}, \boldsymbol{\theta}_q). \tag{3.31}$$

Let us emphasize that the equation of local mechanical equilibrium depends on the parameter $\boldsymbol{\theta}_q$. For the reduced-order model, we would like to have a similar algebraic structure to (3.30),(3.31) but formulated as a low-dimensional system. If $\{\boldsymbol{u}^i\}$ and $\{\boldsymbol{v}^i\}$ are searched in the form $\{\boldsymbol{u}^i\} = \tilde{\Phi}(\boldsymbol{\theta}_q)\,\boldsymbol{\alpha}^i$ and $\{\boldsymbol{v}^i\} = \Gamma(\boldsymbol{\theta}_q)\,\boldsymbol{\xi}^i$, respectively, equation (3.30) becomes

$$\tilde{\Phi}(\boldsymbol{\theta}_q)\,\boldsymbol{\alpha}^{i+1} = \tilde{\Phi}(\boldsymbol{\theta}_q)\,\boldsymbol{\alpha}^i + \delta t^{ROM}\,\Gamma(\boldsymbol{\theta}_q)\,\boldsymbol{\xi}^i.$$

By multiplying by $[\tilde{\Phi}(\boldsymbol{\theta}_q)]^T$ on the left, we get the system of m_u equations

$$\boldsymbol{\alpha}^{i+1} = \boldsymbol{\alpha}^i + \delta t^{ROM}\,Q(\boldsymbol{\theta}_q)\,\boldsymbol{\xi}^i, \tag{3.32}$$

where $Q(\boldsymbol{\theta}_q) = [\tilde{\Phi}(\boldsymbol{\theta}_q)]^T\Gamma(\boldsymbol{\theta}_q)$. Equation (3.31) is replaced by

$$\Gamma(\boldsymbol{\theta}_q)\,\boldsymbol{\xi}^{i+1} = \{\boldsymbol{\varphi}\}(\tilde{\Phi}(\boldsymbol{\theta}_q)\,\boldsymbol{\alpha}^{i+1}, \boldsymbol{\theta}_q).$$

By multiplying by $[\Gamma(\boldsymbol{\theta}_q)]^T$ on the left, we get

$$\boldsymbol{\xi}^{i+1} = \mathcal{M}(\boldsymbol{\alpha}^{i+1}, \boldsymbol{\theta}_q)$$

where

$$\mathcal{M}(\boldsymbol{\alpha}^{i+1}, \boldsymbol{\theta}_q) = [\Gamma(\boldsymbol{\theta}_q)]^T \{\boldsymbol{\varphi}\}(\tilde{\Phi}(\boldsymbol{\theta}_q)\, \boldsymbol{\alpha}^{i+1}, \boldsymbol{\theta}_q). \tag{3.33}$$

3.4.4 Manifold learning

Because of nonlinear terms, the direct computation of $\mathcal{M}(\boldsymbol{\alpha}^{i+1}, \boldsymbol{\theta}_q)$ in (3.33) requires high-dimensional computations, which makes the ROM irrelevant from a performance point of view. To "identify" a low-order manifold $\mathcal{M}$, we rather adopt a data-driven approach based once again of diffuse approximation. We link the entry data $\alpha_k^D(t^i)$, $k = 1, ..., m_u$, $i = 1, ..., N_t$ to the output data $\xi_k^D(t^i)$, $k = 1, ..., m_v$, $i = 1, ..., N_t$ (*'D'* stands for *'data'*). For that, one can compute the orthogonal projections of the pseudo-snapshots over the POD bases, leading to the formulas

$$\alpha_k^D(t^i) = \langle \{\boldsymbol{u}\}(\{\boldsymbol{X}\}, t^i, \boldsymbol{\theta}_q),\ \{\varphi^k\}(\boldsymbol{\theta}_q)\rangle$$

and

$$\xi^D(t^i) = \langle \{\boldsymbol{v}\}(\{\boldsymbol{X}\}, t^i, \boldsymbol{\theta}_q),\ \{\gamma^k\}(\boldsymbol{\theta}_q)\rangle$$

at instants $t^i = i\Delta t$. Manifold learning consists in achieving a (nonlinear) regression method that links entry and output data. We are looking for a manifold representation $\boldsymbol{\xi} = \mathcal{M}(\boldsymbol{\alpha}, \boldsymbol{\theta}_q)$ in the form

$$\xi_k = \boldsymbol{p}(\boldsymbol{\alpha})^T \boldsymbol{a}^k, \quad k = 1, ..., m_v \tag{3.34}$$

where $\boldsymbol{p}(\boldsymbol{\alpha})$ is the vector made of monomials in $\boldsymbol{\alpha}$ of order zero and one, and $\boldsymbol{a}^k \in \mathbb{R}^{m_u+1}$ is a vector of coefficients to be determined from the data. This corresponds to a local linear embedding process. For each $k = 1, ..., m_v$, one looks for a coefficient vector $\boldsymbol{a}^k(t) \in \mathbb{R}^{m_u+1}$ solution of the weighted least square problem

$$\boldsymbol{a}^k(t) = \arg \min_{\boldsymbol{a} \in \mathbb{R}^{m_u+1}} \quad \frac{1}{2} \sum_{i=1}^{N_t} w\left(\frac{|t - t^i|}{R}\right) \left(\boldsymbol{p}(\boldsymbol{\alpha}^D(t^i))^T \boldsymbol{a} - \xi_k^D(t^i)\right)^2 \tag{3.35}$$

where $t \in [0, T_f]$, $w = w(d)$ is the weight function defined in Figure 3.3b and $d = \frac{|t-t^i|}{R}$. This returns a regression function

$$\xi_k = \xi_k(t, \boldsymbol{\alpha}(t)) = \boldsymbol{p}(\boldsymbol{\alpha}(t))^T\, \boldsymbol{a}^k(t). \tag{3.36}$$

3.4.5 Low-order dynamical reduced order model

The resulting time-discrete reduced-order model is then

$$t^{i+1,ROM} = t^{i,ROM} + \delta t^{ROM}, \tag{3.37}$$

$$\boldsymbol{\alpha}^{i+1} = \boldsymbol{\alpha}^{i} + \delta t^{ROM}\, Q(\boldsymbol{\theta}_q)\, \boldsymbol{\xi}^{i}, \tag{3.38}$$

$$\xi_k^{i+1} = \boldsymbol{p}(\boldsymbol{\alpha}^{i+1})^T\, \boldsymbol{a}^k(t^{i+1,ROM}) \quad \forall k \in \{1, ..., m_v\}. \tag{3.39}$$

High-dimensional displacement and velocity fields can be reconstructed as follows:

$$\{\boldsymbol{u}\}\left(\{\boldsymbol{X}\}, t^{i+1,ROM}, \boldsymbol{\theta}_q\right) = \tilde{\Phi}(\boldsymbol{\theta}_q)\, \boldsymbol{\alpha}^{i+1}, \quad \{\boldsymbol{v}\}\left(\{\boldsymbol{X}\}, t^{i+1,ROM}, \boldsymbol{\theta}_q\right) = \Gamma(\boldsymbol{\theta}_q)\, \boldsymbol{\xi}^{i+1}.$$

The online stage of the reduced-order model is summarized in Algorithm 2.

Algorithm 2 Online phase

```
Require: choose a query parameter $\boldsymbol{\theta}_q$, choose a time step $\delta t^{ROM} > 0$.
  Initialization: $t = t^{0,ROM} = 0$, $\boldsymbol{\alpha}^0 = 0$, $\boldsymbol{\xi}^0 = \boldsymbol{\xi}^D(0)$;
  Compute $\Phi(\boldsymbol{\theta}_q)$ from the diffuse approximation approach;
  for $i = 1 \ldots, N_t$ do
    $\{\boldsymbol{u}\}(\{\boldsymbol{x}\}, t^i, \boldsymbol{\theta}_q) \leftarrow \Phi_u^r\, A(t^i)\, \Psi_u(\boldsymbol{\theta}_q)$;
    $\{\boldsymbol{v}\}(\{\boldsymbol{x}\}, t^i, \boldsymbol{\theta}_q) \leftarrow \Phi_v^r\, B(t^i)\, \Psi_v(\boldsymbol{\theta}_q)$;
  end for
  $\mathcal{U}(\boldsymbol{\theta}_q) \leftarrow [\{\boldsymbol{u}\}(\{\boldsymbol{x}\}, t^1, \boldsymbol{\theta}_q), \ldots, \{\boldsymbol{u}\}(\{\boldsymbol{x}\}, t^{N_t}, \boldsymbol{\theta}_q)]$;
  $\mathcal{V}(\boldsymbol{\theta}_q) \leftarrow [\{\boldsymbol{v}\}(\{\boldsymbol{x}\}, t^1, \boldsymbol{\theta}_q), \ldots, \{\boldsymbol{v}\}(\{\boldsymbol{x}\}, t^{N_t}, \boldsymbol{\theta}_q)]$;
  Compute $\tilde{\Phi}(\boldsymbol{\theta}_q)$, $\Gamma(\boldsymbol{\theta}_q)$, $Q(\boldsymbol{\theta}_q)$, $\boldsymbol{\alpha}^D(t^i)$ and $\boldsymbol{\xi}^D(\boldsymbol{\theta}_i)$, $i = 1, ..., N_t$;
  while $t < T_f$ do
    $t \leftarrow t + \delta t^{ROM}$; $t^{i+1,ROM} = t^{i,ROM} + \delta t^{ROM}$;
    $\boldsymbol{\alpha}^{i+1} = \boldsymbol{\alpha}^i + \delta t^{ROM}\, Q(\boldsymbol{\theta}_q)\boldsymbol{\xi}^i$;
    Compute $\boldsymbol{a}^k(t^{i+1,ROM})$, $k = 1, ..., m_v$ from the diffuse approximation approach;
    $\xi_k^{i+1} = \boldsymbol{p}(\boldsymbol{\alpha}^{i+1})^T\, \boldsymbol{a}^k(t^{i+1,ROM})$;
    If needed, reconstruct the high-dimensional displacements/velocity fields:
    $\{\boldsymbol{u}\}\left(\{\boldsymbol{X}\}, t^{i+1,ROM}, \boldsymbol{\theta}_q\right) = \tilde{\Phi}(\boldsymbol{\theta}_q)\, \boldsymbol{\alpha}^{i+1}$;
    $\{\boldsymbol{v}\}\left(\{\boldsymbol{X}\}, t^{i+1,ROM}, \boldsymbol{\theta}_q\right) = \Gamma(\boldsymbol{\theta}_q)\, \boldsymbol{\xi}^{i+1}$;
  end while
```

We give below figure 3.4 the execution structure of the main steps of the prediction model.

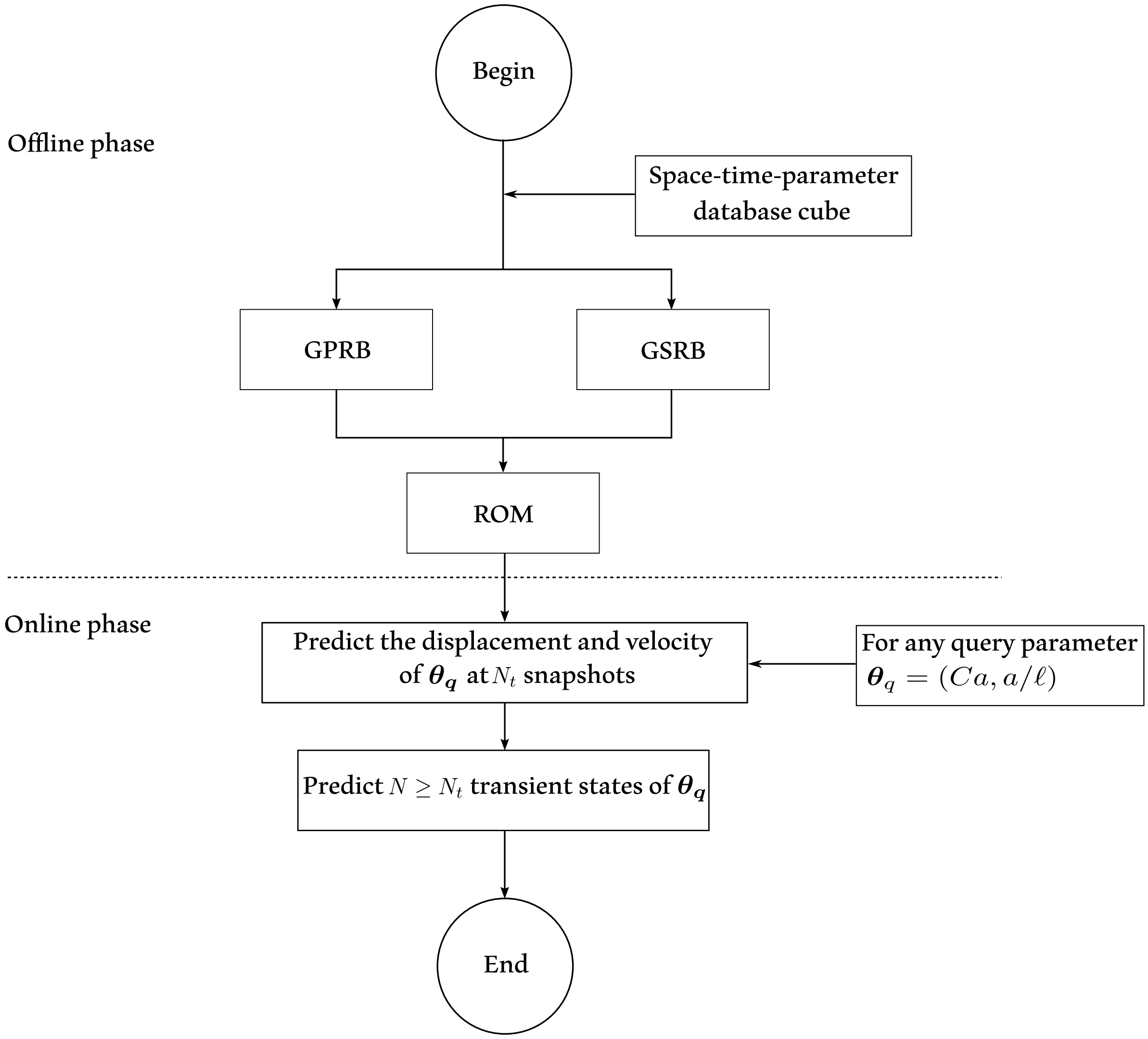


Figure 3.4: Illustration of the main step in the prediction model.

4 Numerical results

4.1 Study case

We consider a capsule flowing in a square-base microchannel of base edges of length 2ℓ. We want to capture the capsule dynamics for capillary numbers Ca belonging to the interval [0.005, 0.2] and aspect ratios a/ℓ in the interval [0.75, 1.2] for which a steady state shape is reached. The Caps3D code (Walter *et al.*, 2010; Hu *et al.*, 2012) is then used as FOM solver. The total non-dimensional time for simulation is $T = 20$. For any capillary number and aspect ratio, the capsule is discretized with the same mesh resolution and connectivity, consisting of $N_x = 2562$ nodes (corresponding to 1280 triangular elements), with a capsule mesh size $\Delta h_C = 0.075\,a$ (see Figure 4.1). A second-order RK2 Ralston scheme is used for time integration. The dimensionless time step is $\dot{\gamma}\delta t = 5\,10^{-4}$ for $Ca > 0.01$ and $\dot{\gamma}\delta t = 10^{-4}$ for $Ca \leq 0.01$.

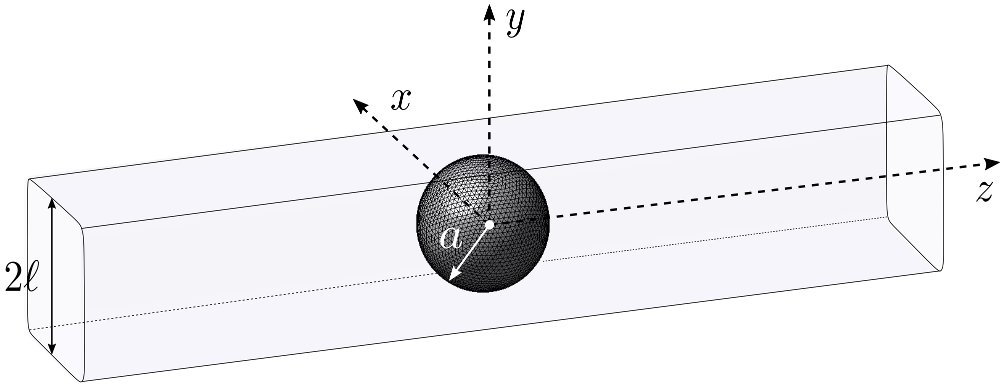


Figure 4.1: Three-dimensional representation of a capsule flowing in a square microchannel.

4.2 FOM result database generation

A database of FOM results is generated from a sampling of the parameter domain (see Figure 4.2). It is observed that configurations for which a shape steady state is reached before the non-dimensional final time of 20 correspond to couples $(Ca, a/\ell)$ in the parameter plane below the dashed red line of Figure 4.2. Using a Cartesian parameter sampling with step sizes of 0.01 in Ca and 0.05 in a/ℓ, plus few additional points at $Ca = 0.005$, we get a database made of $N_c = 118$ configurations. From Caps3D FOM solutions, we pick up time-snapshot solutions every time step $\Delta t = 0.2$ in non-dimensional time scale, corresponding to $N_t = 100$. This makes a datacube made of $2 \times 3N_x N_c N_t \approx 1.81\,10^8$ double precision float numbers taking about 1.45 GB of memory.

4.2.0.1 Clustering strategy For the sake of memory storage complexity, we adopt a strategy of data clustering with two weakly-overlapping clusters chosen manually, represented in Figure 4.2. A clustering study has been done in order to identify the best criteria and the optimal number of clusters (see appendix A) For each cluster, a data dimensionality reduction is done following the offline-stage algorithm presented in Section 3. That means that two families of reduced-order models are actually computed. In the online stage, for a new query parameter vector $\boldsymbol{\theta}_q$, one has to determine the cluster of belonging.

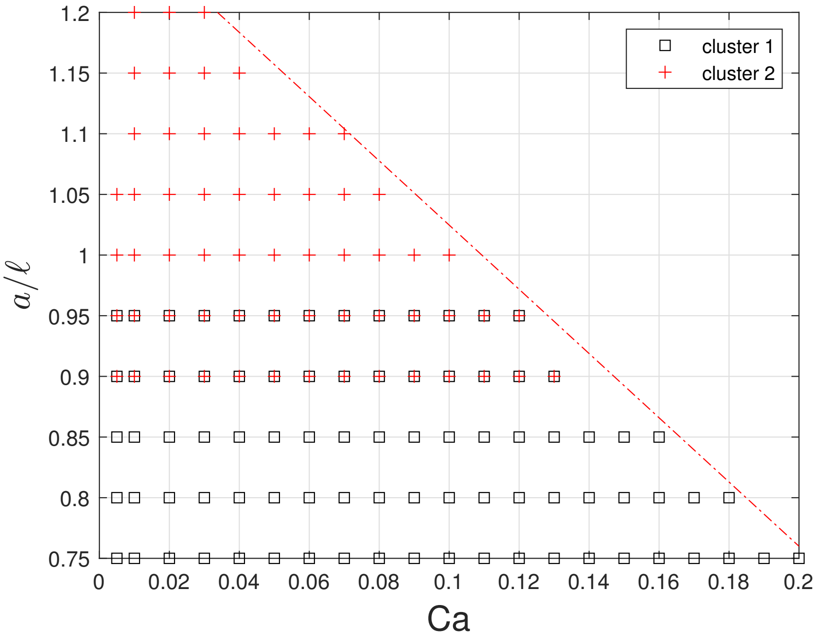


Figure 4.2: Design of computer experiment with sampling in the admissible parameter domain. The parameter domain is divided into two overlapping clusters: cluster 1 (squares), cluster 2 (crosses), and overlapping region (mixed squares and crosses)

4.3 Numerical validation of data-driven reduced-order model

4.3.1 Dimensionality reduction analysis

A singular value decomposition analysis is first performed on the matrices $\mathcal{S}_u$ and $\mathcal{S}_v$, and then on $\mathcal{T}_u$ and $\mathcal{T}_v$. In Figure 4.3(a), we plot the indicator $(1 - \text{RIC})$ (see (6.1)), as a function of the truncation rank K, for $\mathcal{S}_u$ and $\mathcal{S}_v$. What can be seen is that $(1 - \text{RIC})$ rapidly converges towards the value 0 in all cases. An expected $(1 - \text{RIC})$ of 10^{-7} is reached for a truncation rank K_u^c (resp. K_v^c) of 7 for the displacement (resp. 23 for the velocity). Similarly in Figure 4.3(b), we plot the indicator $(1 - \text{RIC})$ for $\mathcal{T}_u$ and $\mathcal{T}_v$. The number of modes K_u^x (resp. K_v^x) needed to reach the threshold of 10^{-7} is 7 for the displacement (resp. 56 for the velocity).

As supplementary indicators, the singular values $\tilde{\sigma}_K$ normalized by $\tilde{\sigma}_1$ are plotted in Figure 4.4(a) (resp. Figure 4.4(b)) for both matrices $\mathcal{S}_u$ and $\mathcal{S}_v$ (resp. $\mathcal{T}_u$ and $\mathcal{T}_v$) in $\log_{10}$ scale. One can first observe a lower decay rate for the velocity fields compared to the displacements, meaning a greater information complexity for the velocity. Secondly, the decay rate is lower for the global spatial mode than for the parametric modes, indicating a larger entropy of information on the whole parameter domain. That justifies the derivation of suitable lower order spatial basis at a query parameter $\boldsymbol{\theta}_q$ in the online stage.

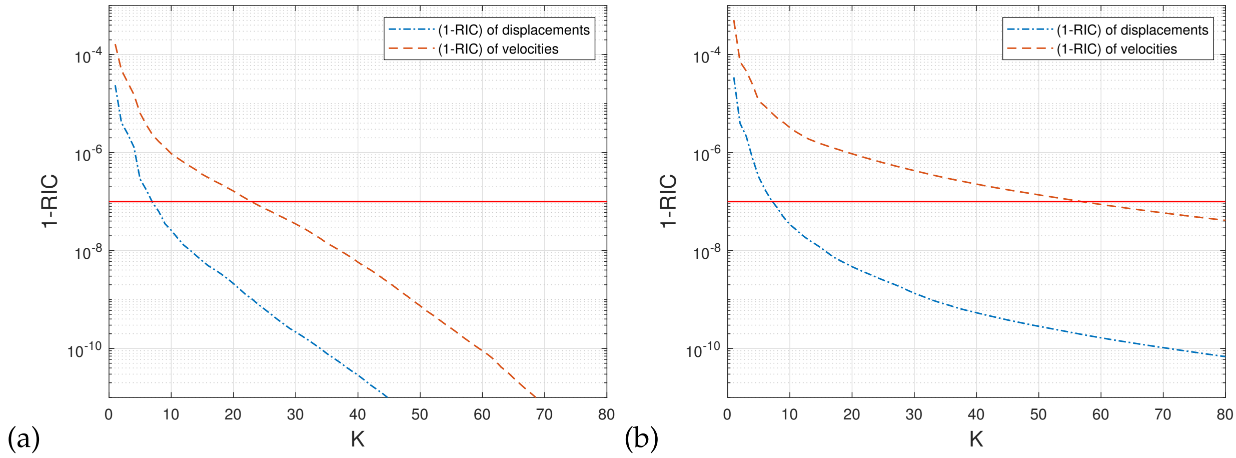


Figure 4.3: behavior of the relative information content of the matrices $\mathcal{S}_u$ and $\mathcal{S}_v$ (a) and $\mathcal{T}_u$ and $\mathcal{T}_v$ (b) is shown in the form $(1 - \mathrm{RIC})$ as a function of the truncation rank K. The red line corresponds to $(1 - \mathrm{RIC}) = 10^{-7}$.

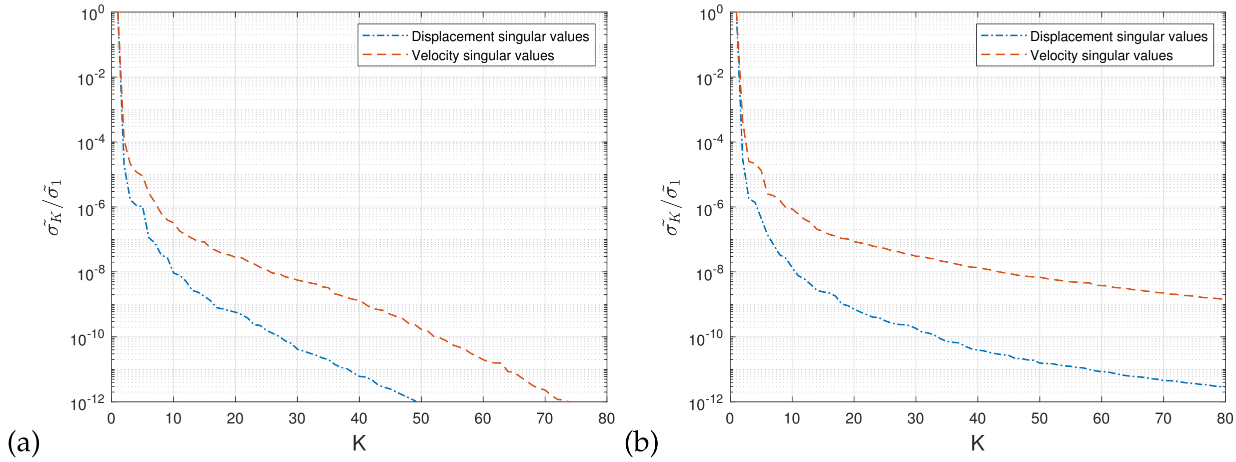
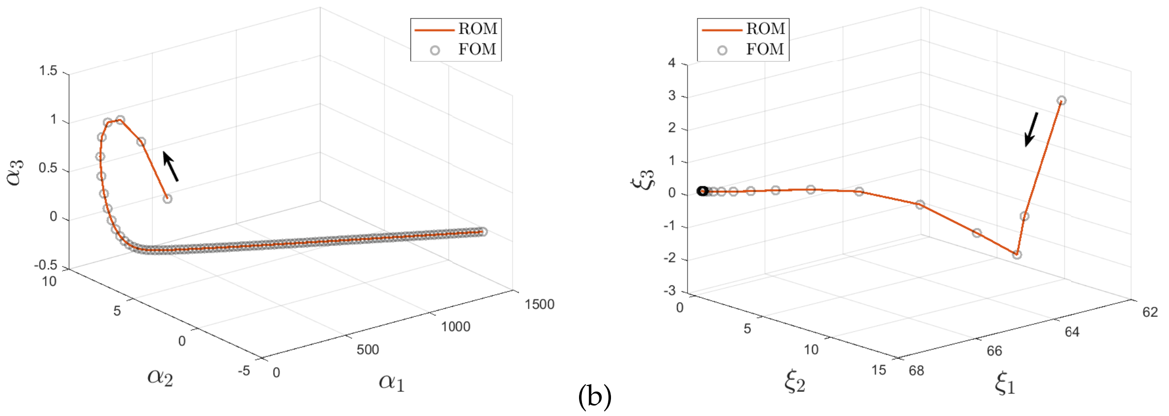


Figure 4.4: (a): Parametric normalized singular values $\tilde{\sigma}_K / \tilde{\sigma}_1$ for $\mathcal{S}_u$ and $\mathcal{S}_v$; (b): Spatial normalized singular values $\tilde{\sigma}_K / \tilde{\sigma}_1$ for $\mathcal{T}_u$ and $\mathcal{T}_v$.

At the beginning of the online stage, for a query parameter $\boldsymbol{\theta}_q$, an interpolated approximate solution is computed thanks to a diffuse approximation reconstruction. This allows us to get pseudo-snapshots in time for both displacements and velocities, stored in matrices $\mathcal{U}(\boldsymbol{\theta}_q)$ and $\mathcal{V}(\boldsymbol{\theta}_q)$, respectively. We assess the RIC for the two matrices, from an experimental parameter vector $\boldsymbol{\theta}_q = (0.10, 0.90)$. The comparison of the time evolution of POD coefficients between FOM and ROM models shows a high accuracy (see Figure 4.5). and Figure 4.6 shows that the RIC rapidly converges to 1. An expected RIC greater than $1 - 10^{-7}$ returns a truncation rank m_u (resp. m_v) of value 3 (resp. 8).



(a) (b)

Figure 4.5: FOM versus ROM comparison of the time evolution of the first three displacements (a) and velocity (b) POD coefficients for the query parameter $\boldsymbol{\theta}_q = (0.10, 0.90)$.

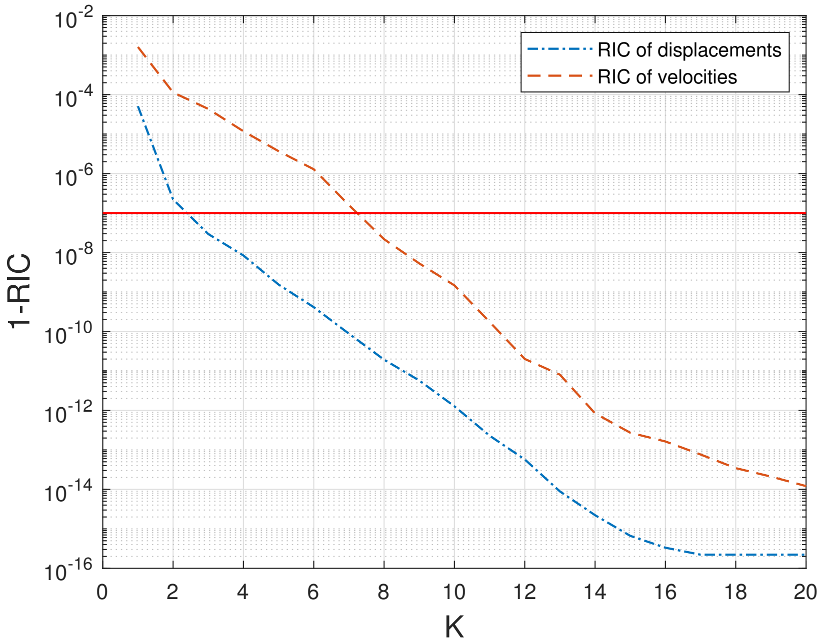


Figure 4.6: Online stage: behavior of the relative information content of the matrices $\mathcal{U}(\boldsymbol{\theta}_q)$ and $\mathcal{V}(\boldsymbol{\theta}_q)$ shown in the form $(1 - \text{RIC})$ for query parameter $\boldsymbol{\theta}_q = (0.10, 0.90)$. The red line corresponds to $(1 - \text{RIC}) = 10^{-7}$.

4.3.2 Offline sensitivity analysis

In this section, we study the effect of the different parameters on the prediction by computing d_{MH}/a between the predicted and high-resolution shapes of a capsule. The study is done using the configuration $\boldsymbol{\theta}_q = (a/\ell = 0.90, Ca = 0.10)$ as an example test which located in cluster 1 (figure 4.2). The default values of the parameters are the following: $(K_u^x, K_u^c) = (40, 75)$,

$(K_v^x, K_v^c) = (40, 75)$, $M = 12$, $(m_u, m_v) = (10, 10)$, and $N_y = 4$.

We start by studying the effect of parametric truncation values K_ϱ^c for $\varrho = u, v$ on the prediction performance. Figures 4.7 shows that 25 parametric modes are enough to reach an error of 0.2% for the displacement model and around 50 modes for the velocity model to reach the same amount of error.

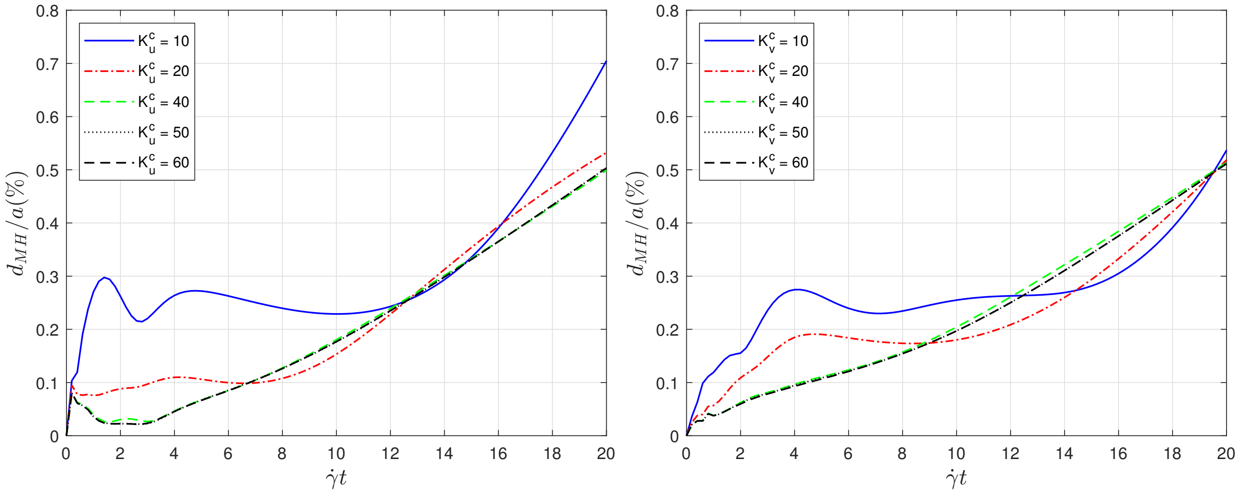


Figure 4.7: The effect of K_ϱ^c parametric truncation parameters on the prediction.

Figures 4.8 shows the effect of spatial truncation parameters on the prediction for different values of K_u^x and K_v^x. As it is shown, the prediction accuracy is proportional to the number of taken modes, and it is relatively identical for displacement and velocity parameters.

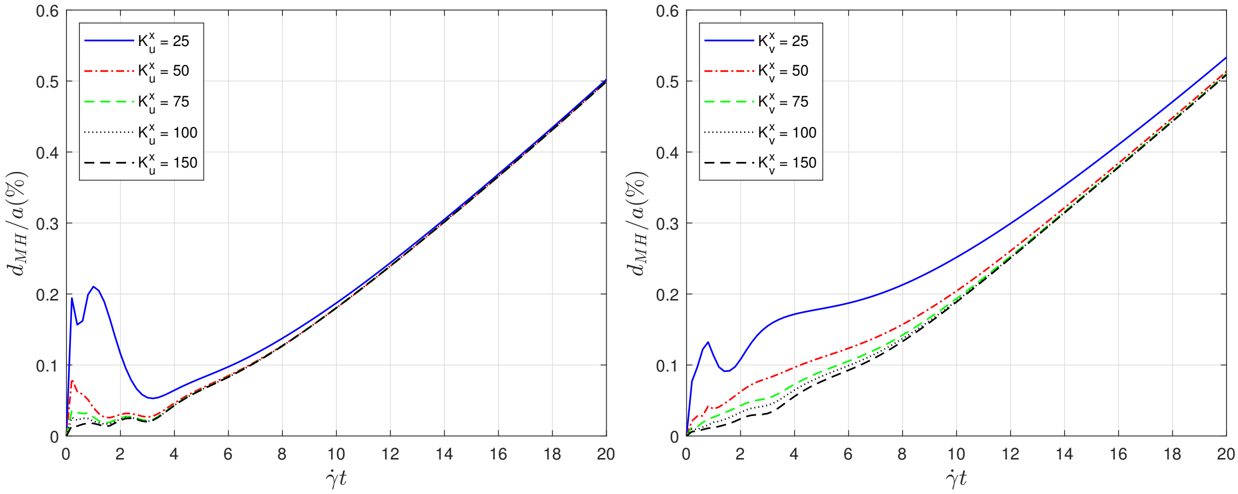


Figure 4.8: The effect of K_ϱ^x parametric truncation parameters on the prediction.

We also study the effect of the number of neighbors that are used in the 2nd order DA method to estimate $\{\boldsymbol{\psi}\}_{q,\varrho}$. Figure 4.9 shows the effect of the number of neighbors on the accuracy of prediction, which is measured by computing normalized Modified Hausdorff distance between high fidelity and reconstructed capsule shapes at several transient states. As the figure shows, the best prediction can be reached when neighbors are under 10 neighbors.

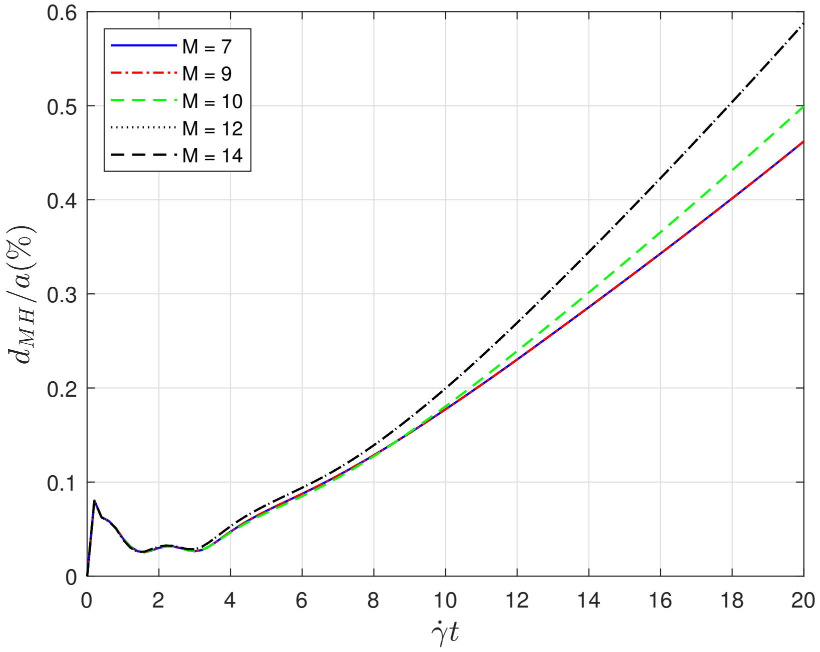


Figure 4.9: The effect of the number of neighbors on the prediction accuracy in the offline phase.

4.3.3 Online sensitivity analysis

At the online phase, we investigate the effect that 3 parameters have on the manifold learning: the truncation parameters for displacement and velocity fields (m_u, m_v) and the number of neighbors for DA method.

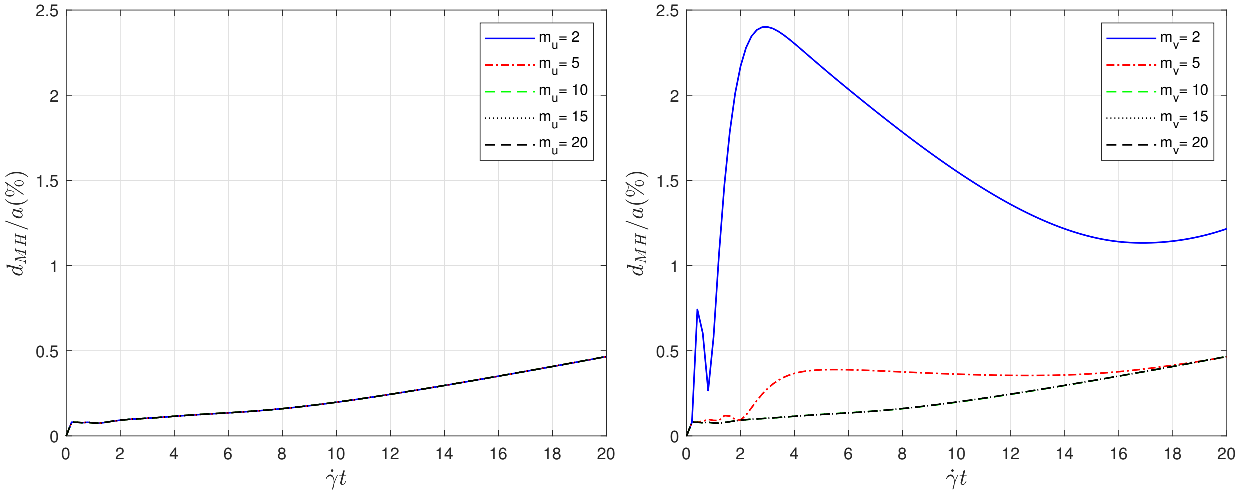


Figure 4.10: The effect of the number of modes on the prediction in the online phase.

Figures 4.10 present the effect of the number of modes m_u and m_v on the prediction where five modes of $\mathcal{U}$ are good enough, unlike $\mathcal{V}$ who needs more modes to reach the best accuracy possible of prediction.

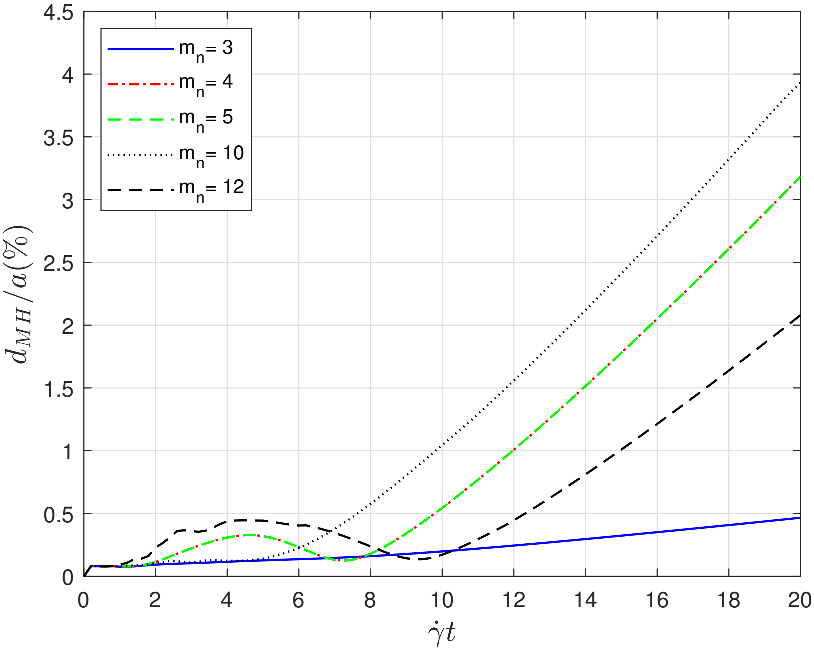


Figure 4.11: The effect of the number of neighbors in the online phase.

Figure 4.11 shows the number of neighbors m_n effect on the prediction where the accuracy of this latter decreases when the number of neighbors rises.

4.3.4 ROM accuracy analysis

The reduced-order model algorithm is applied with the following parameters and options:

- For global POD modes: $K_u^x = 40$, $K_u^c = 40$, $K_v^x = 50$, $K_v^c = 50$;
- For DA in (3.15),(3.20): local second order polynomial reconstruction, $M = 12$;
- For local POD modes: $m_u = 10$, $m_v = 10$;
- For DA in (3.35),(3.36): local first order polynomial reconstruction, $R = 2\Delta t$.

The resulting time-evolution of the three-dimensional capsule shape, that is reconstructed with the ROM model, is illustrated in Figure 4.12 for the query couple $\boldsymbol{\theta}_q = (0.10, 0.90)$. The steady-state is reached before $\dot{\gamma}t = 3$, which explains that the capsule shape is the same for $\dot{\gamma}t = 3, 6, 9$.

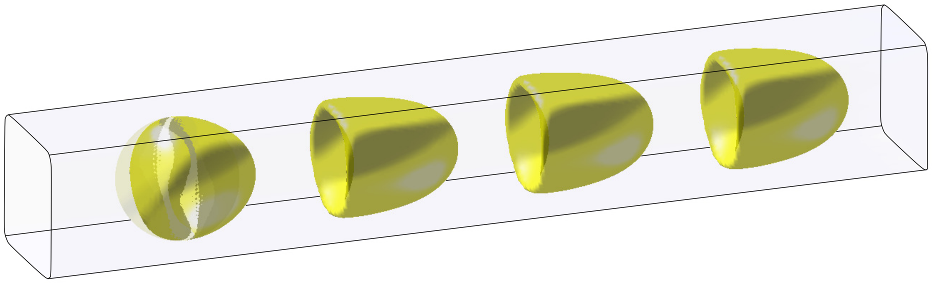


Figure 4.12: A three-dimensional shape of a capsule flowing in a square microchannel, reconstructed with the ROM model for $\boldsymbol{\theta} = (Ca = 0.10, a/\ell = 0.90)$ and shown at $\dot{\gamma}t = 0, 0.4, 3, 6, 9$. The capsule initial shape is shown in transparency.

We now focus on the accuracy analysis of the proposed reduced-order model. The methodology for error measurement is based on a *'Leave-one-out'* cross-validation procedure, where each sample FOM solution is taken out from the database and then evaluated by the ROM model and compared to the original FOM one. The error is measured using the modified Hausdorff distance calculated on the capsule shapes at different instants.

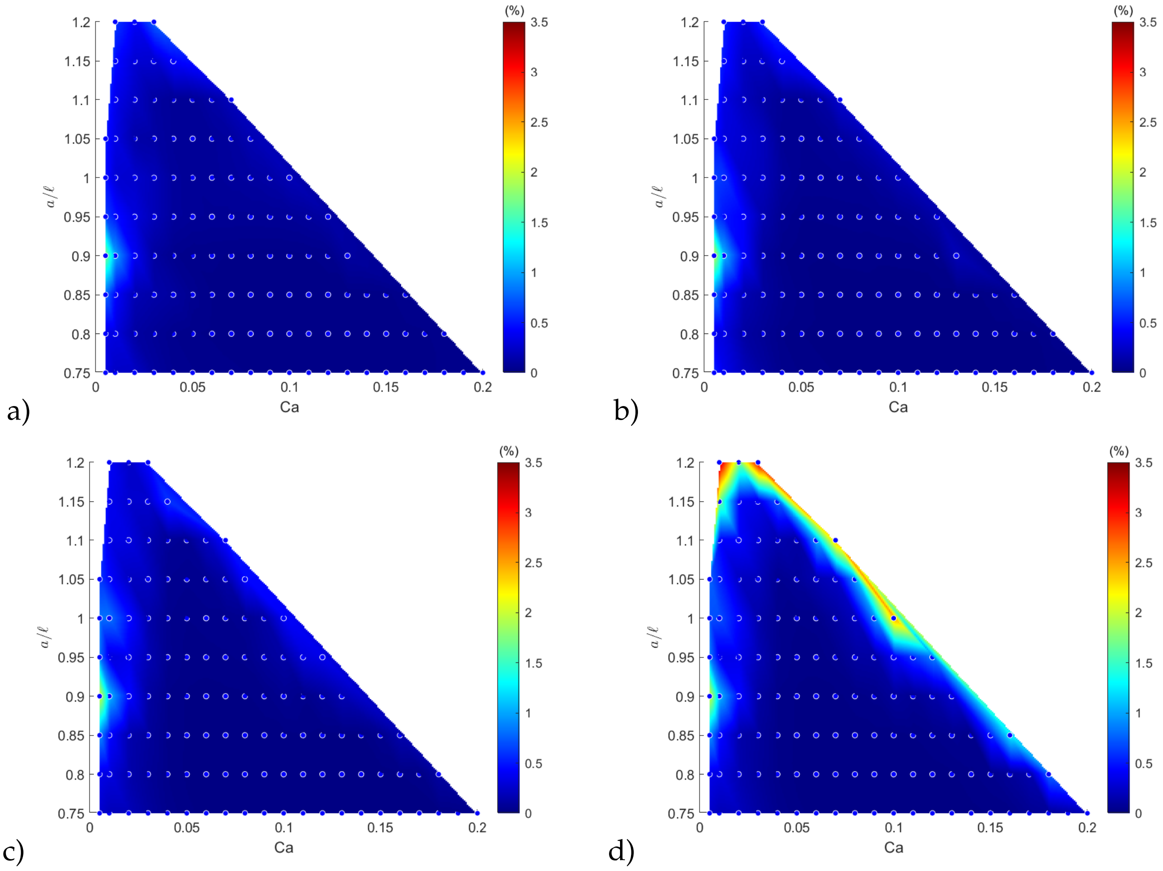


Figure 4.13: Heat maps of the normalized Hausdorff Distance d_{MH}/a of configuration prediction shapes over the parametric space at different transient states: a) $\dot{\gamma}t = 1$; b) $\dot{\gamma}t = 2$; c) $\dot{\gamma}t = 4$; and d) $\dot{\gamma}t = 8$. Note that the maximum error is 3.26% in d).

Figure 4.13 shows the heat maps of the FOM-vs-ROM error computed over the parameter space at the time instants $\dot{\gamma}t = 1, 2, 4$ and 8. Figure 4.13 shows that the predicted ROM solutions are very accurate with a mean relative error below 0.2%. The maximum relative errors are below 3.5%: they occur along the boundary of the parameter domain, which is the only location where the predictions slightly lose in accuracy. This is probably due to a lack of well-distributed neighbors close to the boundaries, which affects the accuracy of the DA reconstruction (off-centre approximation). One can also notice that the accuracy of predictions decreases in time.

The capsule cross-section profiles predicted by the ROM (red dots) are compared to the reference FOM solutions (solid black line) in Figure 4.15 at different time instants ($\dot{\gamma}t = 0$, 1, 2 and 8) for the 6 configurations, selected as illustration in Figure 2.2.

To seek this purpose, we apply a vertical cutting plane on the 3D capsule shape figure 4.14.

We observe that the reduced-order model returns very accurate solutions in terms of capsule shape as well-as axial position in the channel.

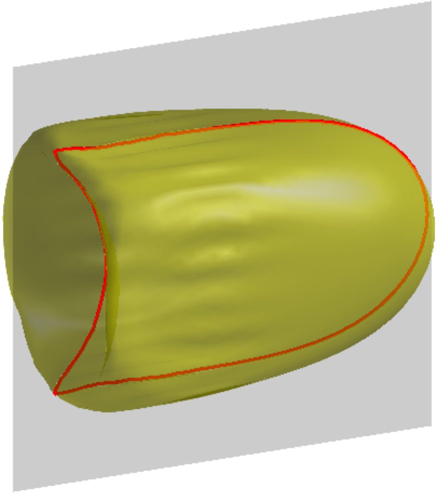


Figure 4.14: A microcapsule of $\boldsymbol{\theta} = (0.068, 1.077)$ flowing in a square microchannel simulated by FOM with a vertical cutting plane represented in grey.

From the computing performance point of view, ROM-vs-FOM speedups are observed to be of order 10,000 with almost the same accuracy, making interactive exploration and real-time visual rendering possible.

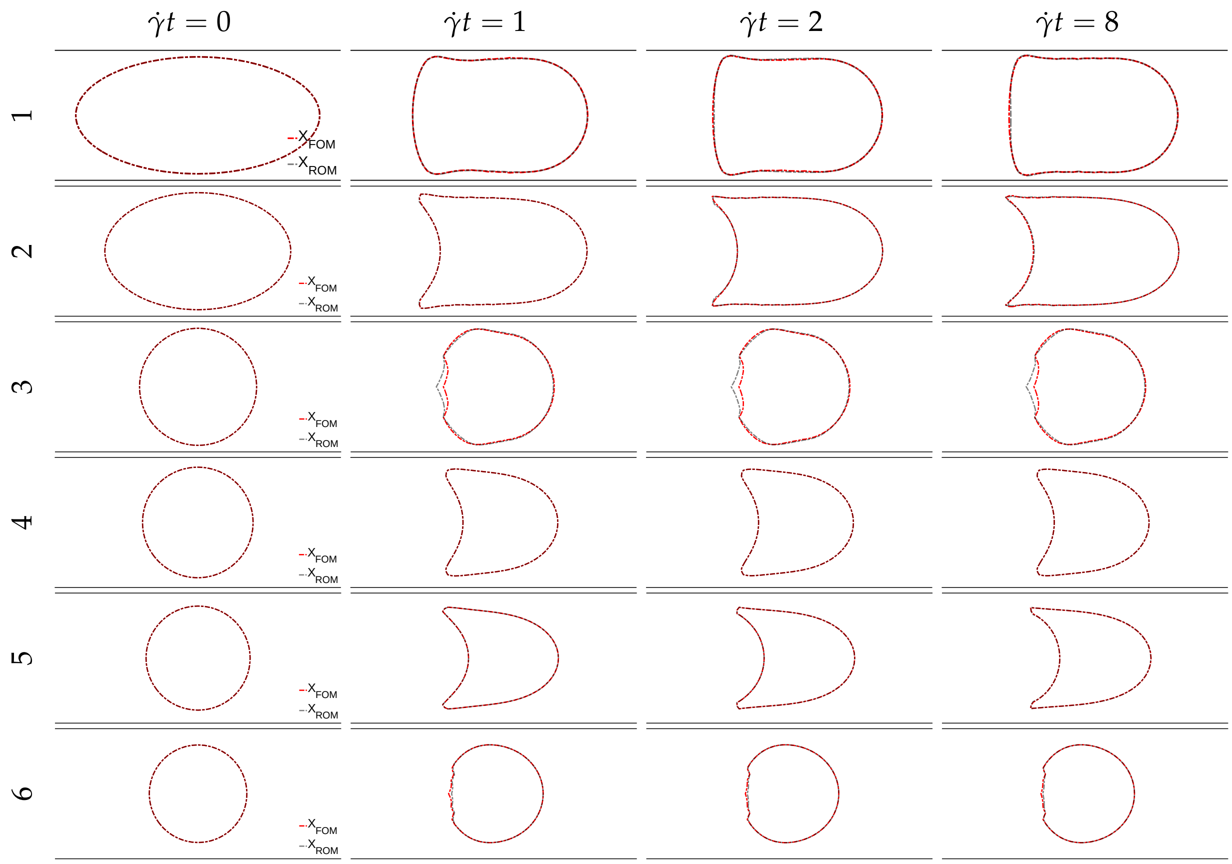


Figure 4.15: Comparison between the ROM (red dots) and FOM solutions (black line) of the capsule cross-section shapes in the plane $y = 0$ at the times $\dot{\gamma}t = 0$, 1, 2, and 8, respectively, for the 6 parameter couples selected in Figure 2.1. The horizontal lines correspond to the channel walls.

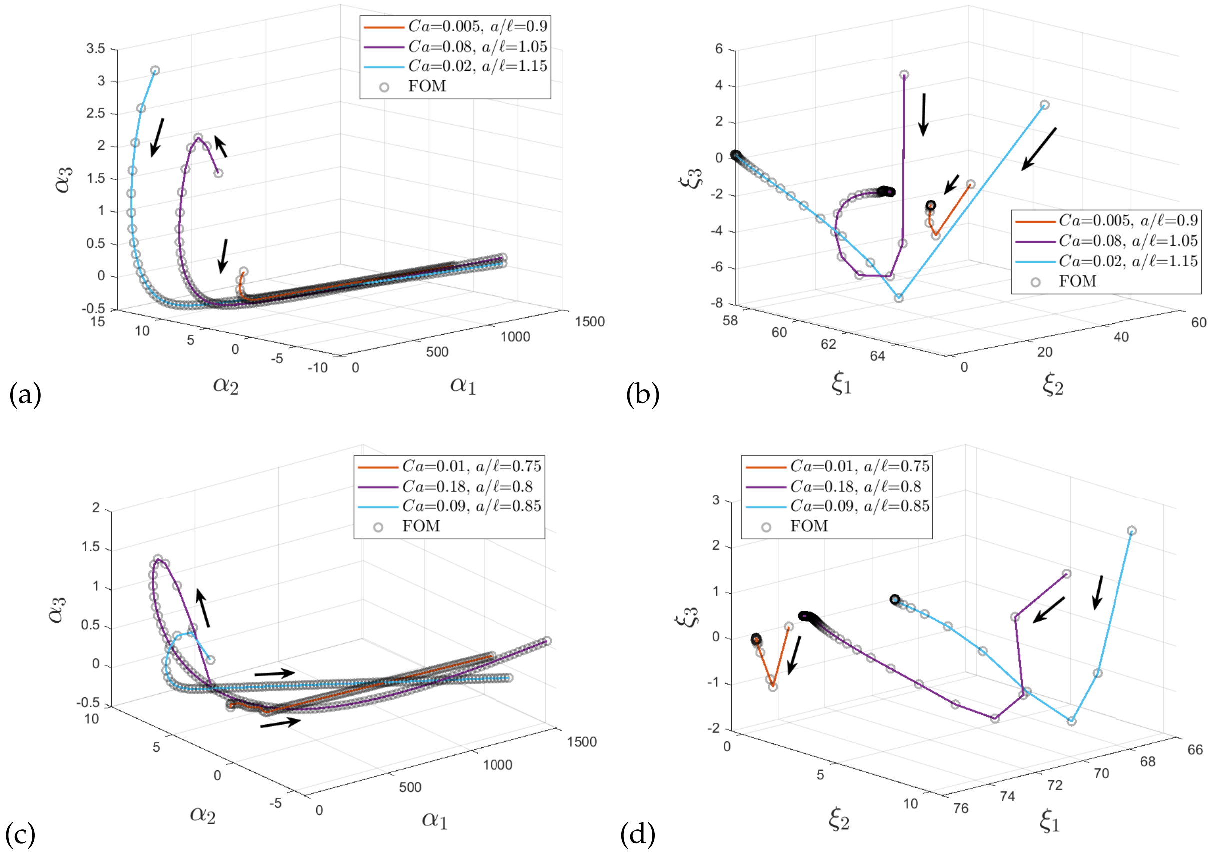
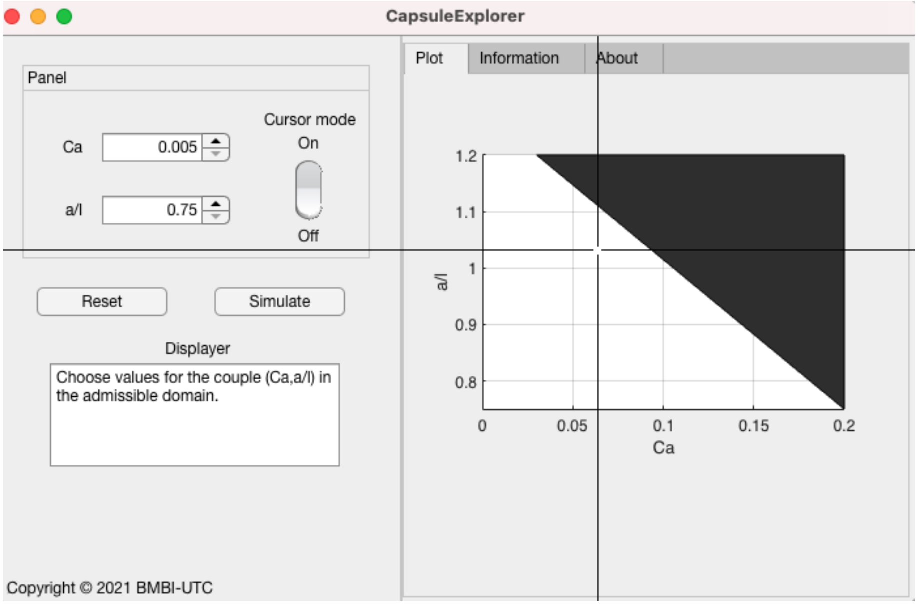


Figure 4.16: Time evolution of the first 3 POD coefficients of the displacement (a,c) and velocity (b,d) for the selected query parameters of Figures 2.2-4.15. The black arrows indicate the time-evolution direction.

5 Software tool presentation

We have developed a software tool `CapsuleExplorer` based on the data-driven ROM to provide 3D microcapsule deformation for any couple $(Ca, a/\ell)$ in the admissible parameter space (figure 2.1) at any time $\dot{\gamma}t$. The tool allows to select a couple $(Ca, a/\ell)$ and explores in real-time its corresponding capsule deformation through time, either in 3D or 2D with longitudinal or transversal cross-sectional view. It uses an optimal experimental design for a faster resolution (see the appendix B). In the following, we explain how to use the software interface.

1. Entering a capillary number Ca and a confinement ratio a/ℓ can be done either using the spinners or by activating the cursor mode switcher and clicking directly with the mouse



Ca 0.005

a/l 0.75

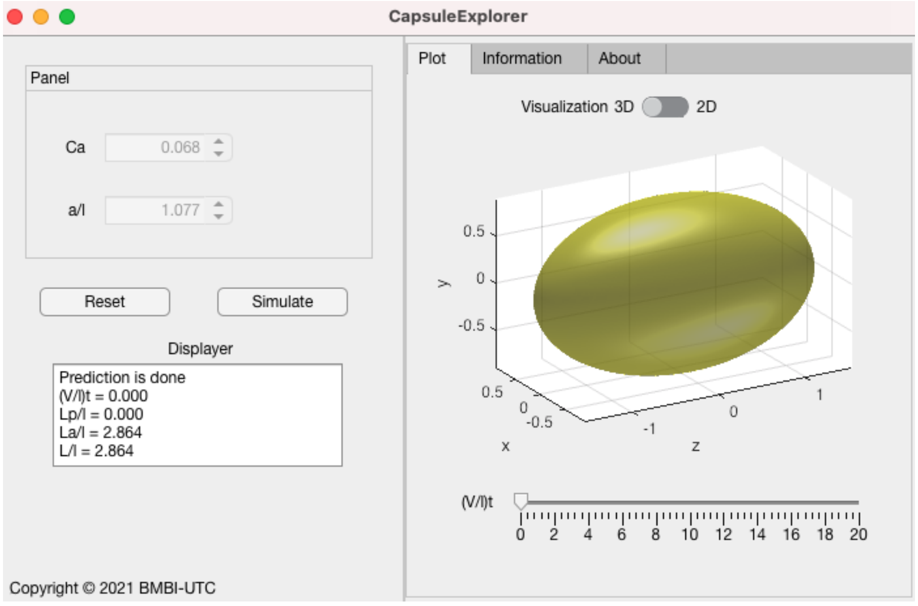
Choose values for the couple (Ca,a/l) in the admissible domain.

Copyright © 2021 BMBI-UTC

Figure 5.1: Selecting capillary number and confinement ratio using cursor mode.

cursor on the white zone shown on the right of figure 5.1. The selected configuration can only be in the white zone, as it is the region of the parametric domain where a capsule obeying the Neo-Hookean law reaches a steady-state shape.

2. Click on "Simulate". Once the “Displayer”, located at the bottom left of the screen, indicates



Ca 0.068

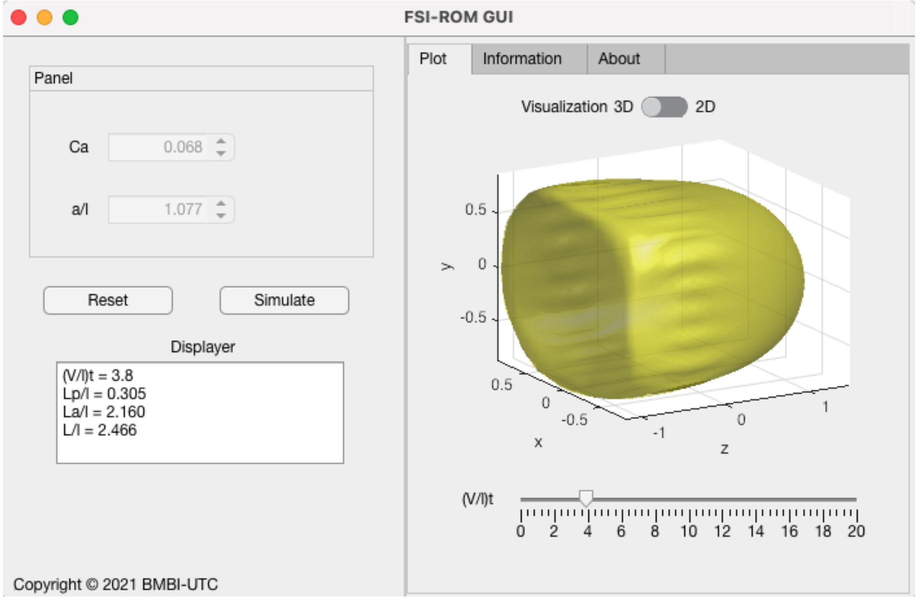
a/l 1.077

Prediction is done
(V/l)t = 0.000
Lp/l = 0.000
La/l = 2.864
L/l = 2.864

Copyright © 2021 BMBI-UTC

Figure 5.2: Initial capsule shape in 3D view.

the end of the calculation, the 3D capsule shape appears automatically on the right (in its initial spherical shape). It can be viewed at any other time using the timer slider located



FSI-ROM GUI

Panel

Ca 0.068

a/l 1.077

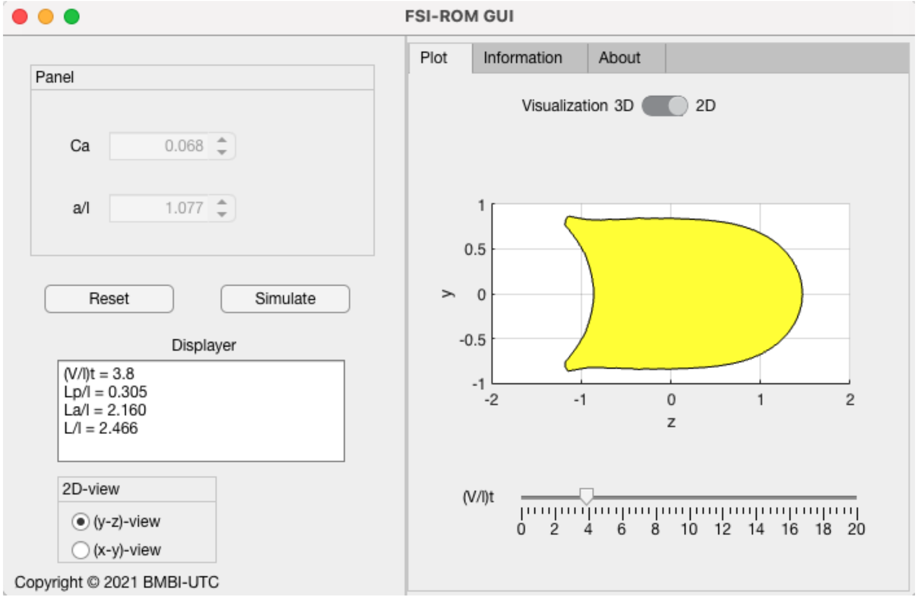
Displayer

(V/l)t = 3.8
Lp/l = 0.305
La/l = 2.160
L/l = 2.466

Copyright © 2021 BMBI-UTC

Figure 5.3: Dynamic 3D capsule view.

below the figure.



FSI-ROM GUI

Panel

Ca 0.068

a/l 1.077

Displayer

(V/l)t = 3.8
Lp/l = 0.305
La/l = 2.160
L/l = 2.466

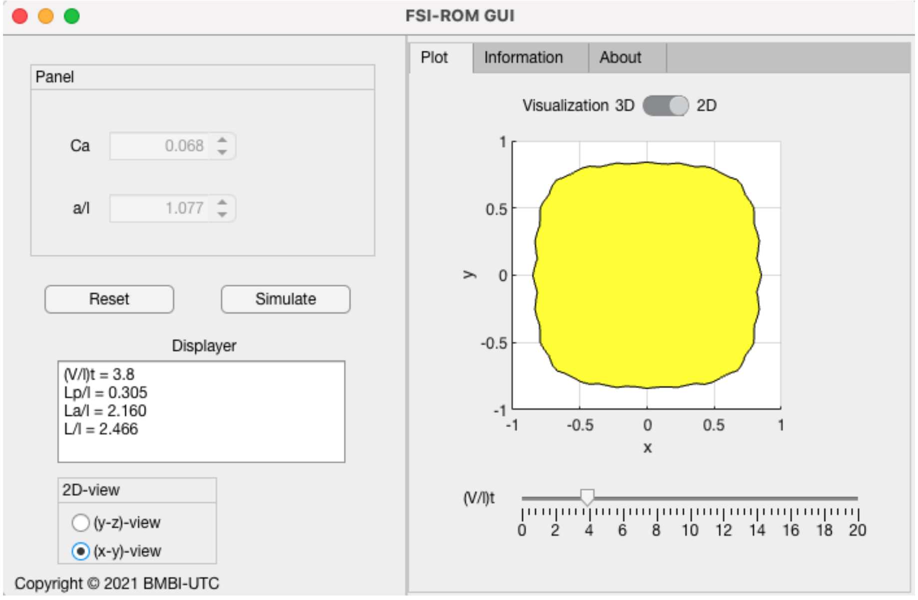
2D-view

● (y-z)-view
○ (x-y)-view

Copyright © 2021 BMBI-UTC

Figure 5.4: Dynamic 2D cross-section longitudinal view.

3. In the 2D view, one can visualize the cross-cut profiles of the capsule by choosing the $(y-z)-$view or its transverse view by selecting the $(x-y)-$view from the radio buttons, located below the “Displayer” (figure 7). As shown in figures 5.4 and 5.5, the "Displayer" provides, at each chosen time instant $(V/\ell)\,t$, the values of the geometrical parameters defined in figure 2.1.



Ca 0.068

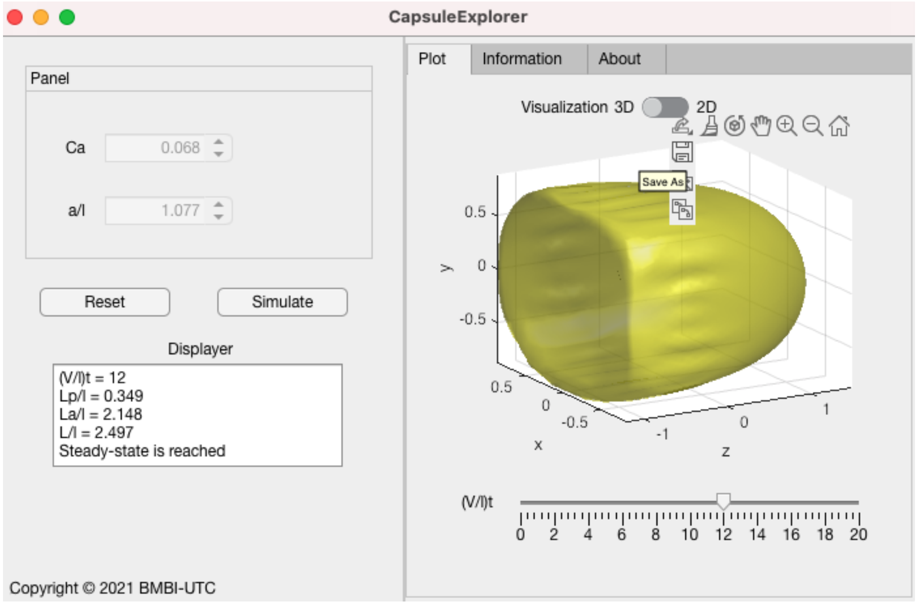
a/l 1.077

(V/l)t = 3.8
Lp/l = 0.305
La/l = 2.160
L/l = 2.466

○ (y-z)-view
● (x-y)-view

Figure 5.5: Dynamic 2D cross-section transversal view.

4. one can also save visulaized capsule shape by selecting "save as" option.



Ca 0.068

a/l 1.077

(V/l)t = 12
Lp/l = 0.349
La/l = 2.148
L/l = 2.497
Steady-state is reached

Figure 5.6: Saving the capsule shape.

5. To launch a new simulation, click on the "Reset" button.

6 Concluding remarks

This chapter presents an innovative data-driven reduced-order model that enables the fast-simulation of capsule deformation flowing in a microfluidic channel. It consists of building a data-driven ROM approach that allows predicting the time evolution of the capsule deformation behavior for any couple of parameters for the admissible parametric space. Numerical validations confirm the efficiency of the two methods. For the ROM model, speedups are observed to be of order $10,000$ compared to FOM, with an average accuracy of 0.3%.

Chapter 5

A non-intrusive kinematics-consistent dynamic reduced-order model

1 Introduction

In this chapter, we propose a dynamical time-space order reduction model for the deformable microcapsules in microfluidic channel. This model uses displacement and velocity snapshot matrices extracted from FOM simulations following the work of De Vuyst *et al.* (2022). The approach combines Proper Orthogonal Decomposition (POD) and Dynamic Mode Decomposition (DMD) and is used to build a dynamical system able to predict the capsule deformability in time. A first model of that kind was recently proposed by Dupont *et al.* (2021) where they showed how accurate POD-DMD reduced order model was to predict the capsule dynamics. The objective is now to investigate what happens when the matrix of the DMD model has non constant coefficient and whether it adds even more precision.

We start the chapter by giving a differential algebraical system that expresses the FSI problem. We formulate then a dynamical system that describes the capsule deformation dynamic through a non constant dynamic operator. This latter is identified in section 3 using the Dynamic Decomposition Method. In section 4, we only show results for the first and the second iterations, as the model crashes at the following one.

2 Differential algebraical system

As we did for the first model, we start from FSI resolution database, where the relationship between the evolution of the capsule displacement $\boldsymbol{u}$ and its velocity $\boldsymbol{v}$ can be described through an Algebraic Differential Equation (DAE)

$$\begin{aligned} \{\dot{u}\} &= \{v\},\ t \in (0,T] \\ \{v\} &= \phi\left(\{u\}\right), \\ \{u\}(0) &= 0, \end{aligned} \tag{2.1}$$

where ϕ is a nonlinear application and with the reconsideration of the velocity $\{v\}$ linear combination formula for any configuration of parameters $\boldsymbol{\theta}$

$$\{v\}(t) \simeq \sum_{k=1}^{K} \xi_k(t)\,\{\gamma\}_k(\boldsymbol{\theta}). \tag{2.2}$$

From kinematic equation 2.1, the integration of the formula (2.2) in time allows us to have a reduced formula of capsule displacement

$$\{u\}(t) = \int_0^t \xi_k(s)\,\{\gamma\}_k(\boldsymbol{\theta})\,ds. \tag{2.3}$$

It can be written in the from

$$\{u\}(t) \simeq \sum_{k=1}^{K} \alpha_k(t)\,\{\gamma\}_k(\boldsymbol{\theta}), \tag{2.4}$$

where $\alpha_k(t) = \int_0^t \xi_k(s)\,ds$. With the Galerkin projection of the system 2.1 on the POD basis, we can rewrite the DAE system as

$$\begin{aligned} \dot{\boldsymbol{\alpha}}(t) &= \boldsymbol{\xi}(t), \\ \xi_k &= \langle \{\gamma\}_k, \phi\left(\textstyle\sum_{k=1}^{K} \alpha_k(t)\,\{\gamma\}_k\right)\rangle,\ k = 1,\dots,K. \end{aligned} \tag{2.5}$$

it can also be written implicitly in the form

$$\begin{aligned} &\dot{\boldsymbol{\alpha}}(t) = \boldsymbol{\xi}(t), \\ &\boldsymbol{\xi}(t) = \phi_r\left(\boldsymbol{\alpha}(t)\right), \qquad \text{with } \boldsymbol{\alpha}(0) = 0, \end{aligned} \tag{2.6}$$

where ϕ_r is an unknown nonlinear function in $\mathbb{R}^K$. The evaluation of ϕ_r usually requires high-dimensional operation which making the approach losing its dimensional reduction benefit. Alternatively, it can be identified using FOM simulation results. For that, the reformulation of DAE system to an Ordinary Differential Equation (ODE) is proposed in the next section.

2.1 ODE system transition

The DAE system (2.1) can be transformed to an Ordinary Differential Equation (ODE) system, as follows:

$$\begin{aligned} \dot{\boldsymbol{\alpha}}(t) &= \boldsymbol{\xi}(t), \\ \dot{\boldsymbol{\xi}}(t) &= \left(\frac{\partial \phi_r}{\partial \alpha}\right)_{\boldsymbol{\alpha}=\boldsymbol{\alpha}(t)} \dot{\boldsymbol{\alpha}}(t) = \left(\frac{\partial \phi_r}{\partial \alpha}\right)_{\boldsymbol{\alpha}=\boldsymbol{\alpha}(t)} \boldsymbol{\xi}(t), \end{aligned} \tag{2.7}$$

where $\left(\frac{\partial \phi_r}{\partial \alpha}\right) \dot{\boldsymbol{\alpha}}$ is the jacobian matrix of $\phi_r \in \mathbb{R}^{K\times K}$ at the point $\boldsymbol{\alpha}$. For simplification reason we note:

$$\boldsymbol{A}(\boldsymbol{\alpha}) = \left(\frac{\partial \phi_r}{\partial \alpha}\right)(\dot{\boldsymbol{\alpha}}). \tag{2.8}$$

By replacing (2.8) in (2.7), we obtain the ODE system:

$$\begin{aligned} \dot{\boldsymbol{\alpha}}(t) &= \boldsymbol{\xi}(t), \\ \dot{\boldsymbol{\xi}}(t) &= \boldsymbol{A}(\boldsymbol{\alpha})\,\boldsymbol{\xi}(t) \end{aligned} \tag{2.9}$$

with the limite conditions $\xi_k(0) = \langle \{\gamma\}_k, \phi(\{\boldsymbol{X}\})\rangle$, $k = 1, \ldots, K$.

3 Identification of the dynamic operator

The system 2.9 can be solved if and only if the matrices $\boldsymbol{A}(t)$ are determined. One way to identify the matrices is by minimizing the least square functional

$$\min_{A(t)\in\mathcal{M}_K(\mathbb{R})} \frac{1}{2}\int_0^T \left\| \dot{\boldsymbol{\xi}}(t) - \boldsymbol{A}(t)\,\boldsymbol{\xi}(t) \right\|^2 dt, \tag{3.1}$$

where a simpler case in which $\boldsymbol{A}(t)$ can be determined using 3.1 is when it is considered as a constant matrix $\boldsymbol{A}$.

3.1 Identification of A (α) as a constant matrix

From the FOM database, the identification of the constant matrix **A** can be done using a discretization in time $t^n = n\Delta t$ where Δt is the time step which leads to write:

$$\min_{A\in\mathcal{M}_K(\mathbb{R})} \frac{1}{2}\sum_{n=1}^{N-1} \left\| \frac{\boldsymbol{\xi}^{n+1} - \boldsymbol{\xi}^n}{\Delta t} - \boldsymbol{A}\boldsymbol{\xi}^n \right\|^2, \tag{3.2}$$

where $\frac{\boldsymbol{\xi}^{n+1}-\boldsymbol{\xi}^n}{\Delta t}$ is a forward the finite difference approximation of $\dot{\boldsymbol{\xi}}$ at time t^n. The minimization problem 3.2 can be written in condensed matrix form

$$\min_{A\in\mathcal{M}_K(\mathbb{R})} \frac{1}{2}\|\mathbb{Y} - \boldsymbol{A}\mathbb{X}\|_F^2, \tag{3.3}$$

where $\|\cdot\|_F$ is the Frobenius norm and the two data matrices

$$\mathbb{X} = \left[\frac{\boldsymbol{\xi}^2 - \boldsymbol{\xi}^1}{\Delta t}, \ldots, \frac{\boldsymbol{\xi}^N - \boldsymbol{\xi}^{N-1}}{\Delta t}\right], \; \mathbb{Y} = \left[\boldsymbol{\xi}^1, \ldots, \boldsymbol{\xi}^{N-1}\right]. \tag{3.4}$$

The solution of the quadratic problem is given by

$$A = \mathbb{Y}\mathbb{X}^{\dagger}, \tag{3.5}$$

where $\mathbb{X}^{\dagger} = \mathbb{X}\left(\mathbb{X}\mathbb{X}^T\right)^{-1}$ is the pseudo-inverse matrix. With the identification of A, the system (2.9) can be resolved through a numerical scheme as an explicit Euler scheme which gives

$$\begin{aligned} \boldsymbol{\alpha}^{n+1} &= \boldsymbol{\alpha}^{n} + \Delta t \boldsymbol{\xi}^{n}, \\ \boldsymbol{\xi}^{n+1} &= \boldsymbol{\xi}^{n} + \Delta t A \boldsymbol{\xi}^{n}. \end{aligned} \tag{3.6}$$

The stability of the discrete system depends on the spectral proprieties of A. A regularization of A can be used to get a more robust system by adding a Tikhonov regularization term

$$\min_{A\in\mathcal{M}_K(\mathbb{R})} \frac{1}{2}\|\mathbb{Y} - A\mathbb{X}\|_F^2 + \frac{\mu}{2}\|\mathbb{X}\|_F^2\|A\|_F^2, \tag{3.7}$$

where the scalar $\mu > 0$ is the regularization coefficient (Dupont *et al.*, 2021)). The solution A_μ of the formula 3.7 is given by

$$A_\mu = \mathbb{Y}\mathbb{X}^T\left(\mathbb{X}\mathbb{X}^T + \mu\|\mathbb{X}\|_F^2 I_K\right)^{-1}. \tag{3.8}$$

Since the matrix A_μ depends on the regularization coefficient μ, the optimal choice of this latter should be taken by considering the approximation quality estimated by $\|\mathbb{Y} - A_\mu\mathbb{X}\|_F^2$ and the norm solution $\|A_\mu\|_F^2$. The optimal value of μ is located at the corner of the $L-$curve that is plottes the values of $\mu \rightarrow \|\mathbb{Y} - A_\mu\mathbb{X}\|_F^2$ according to the residual $\mu \rightarrow \|A_\mu\|_F^2$.

3.2 Identification of A $(\boldsymbol{\alpha})$ with DMD approach

To build a dynamic data-driven system from the system 2.9, one can decompose the matrix $A(\boldsymbol{\alpha})$ under a dense base. For that, let $\left(E_{i,j}\right)_{i,j=1,\dots,K}$ the canonical base of $\mathcal{M}_K(\mathbb{R})$, and $a_{ij}(\boldsymbol{\alpha}) \in \mathbb{R}$ an elements of $A(\boldsymbol{\alpha})$, so we have

$$A(\boldsymbol{\alpha}) = \sum_{i=1}^{K}\sum_{j=1}^{K} a_{ij}(\boldsymbol{\alpha}) E_{ij} \tag{3.9}$$

and by considering the all functions $\boldsymbol{\alpha} \rightarrow a_{ij}(\boldsymbol{\alpha})$ belong to a Banach space $\mathcal{E}$. Let $(\boldsymbol{\alpha} \rightarrow \phi_k(\boldsymbol{\alpha}))_{k\geq 0}$ a dense family in $\mathcal{E}$. Now, by assuming that each mapping $a_{ij}(\boldsymbol{\alpha})$ acquires the decomposition

$$a_{ij}(\boldsymbol{\alpha}) = \sum_{k\geq 0} a_{ij;k}(\boldsymbol{\alpha}). \tag{3.10}$$

We can rewrite (3.9) as

$$A(\boldsymbol{\alpha}) = \sum_{i=1}^{K}\sum_{j=1}^{K}\sum_{k\geq 0} a_{ij;k}(\boldsymbol{\alpha}) E_{ij}. \tag{3.11}$$

The permutation of summation operators in formula (3.11) allows to write the following decomposition

$$\boldsymbol{A}\left(\boldsymbol{\alpha}\right)=\sum_{k\geq 0}\boldsymbol{A}_k\phi_k\left(\boldsymbol{\alpha}\right), \tag{3.12}$$

where the constants matrices

$$\boldsymbol{A}_k=\sum_{i=1}^{K}\sum_{j=1}^{K}a_{ij;k}E_{ij}. \tag{3.13}$$

The $\boldsymbol{A}\left(\boldsymbol{\alpha}\right)$ can be determined knowing $\boldsymbol{A}_k$ matrices and $\phi_k\left(\boldsymbol{\alpha}\right)$ functions which should be chosen. For that, we can propose an iterative algorithm of the presented approach above with considering $\phi_k\left(\boldsymbol{\alpha}\right)=\alpha_k$. This latter represent the features that allow to identify $\boldsymbol{A}\left(\alpha\right)$ through the formula 3.12. There is no trivial way to choose these features. In the literature, there are number of techniques for constructing the feature space (Ham *et al.*, 2004; Cavoretto & De Rossi, 2016). For instance, a simple choice can be done by considering

$$\alpha_0=1, \qquad \alpha_k\left(\boldsymbol{\alpha}\right)=\boldsymbol{\alpha}\cdot\boldsymbol{e}_k=\alpha_k, \quad \text{for } k=1,\ldots,K, \tag{3.14}$$

where $\boldsymbol{e}_k$ are the vectors of the canonic base in $\mathbb{R}^K$ (Erichson *et al.*, 2019). From the two precedent sections, one can suggest an iterative algorithm in which the matrices $\boldsymbol{A}_k$ are computed successively. So, from the formula 3.9 we have

$$\begin{cases}\boldsymbol{A}_k\left(\boldsymbol{\alpha}\right)=\boldsymbol{A}_{(k-1)}\left(\boldsymbol{\alpha}\right)+\alpha_k\boldsymbol{A}_k, & \text{for } k=1,\ldots,K.\\ \boldsymbol{A}_0\left(\boldsymbol{\alpha}\right)=\boldsymbol{A}_0\end{cases} \tag{3.15}$$

where $\boldsymbol{A}_0$ can be computed either with the formulas 3.3 or 3.8. $\boldsymbol{A}_k$ is estimated by minimizing the least square objective function

$$\min_{\boldsymbol{A}_k}\frac{1}{2}\sum_{n=1}^{N-1}\parallel\left(\mathbb{Y}\right)^n-\left(\boldsymbol{A}_{(k-1)}\left(\boldsymbol{\alpha}\right)+\boldsymbol{A}_k\alpha_k^n\right)\left(\mathbb{X}\right)^n\parallel_F^2, \tag{3.16}$$

where $\left(\mathbb{Y}\right)^n$ and $\left(\mathbb{X}\right)^n$ are the $n-$th column of the matrices $\mathbb{Y}$ and $\mathbb{X}$, respectively. Once the matrix $\boldsymbol{A}\left(\alpha\right)$ is identified, the system (2.9) can be resolved through a numerical scheme as explicit Euler scheme which gives

$$\begin{aligned}\boldsymbol{\alpha}^{n+1}&=\boldsymbol{\alpha}^n+\Delta t\boldsymbol{\xi}^n,\\ \boldsymbol{\xi}^{n+1}&=\boldsymbol{\xi}^n+\Delta t\boldsymbol{A}\left(\alpha\right)\boldsymbol{\xi}^n,\end{aligned} \tag{3.17}$$

where the stability properties of the discrete system depend on the spectral properties of the matrix

$$\boldsymbol{A}^{\Delta}\left(\alpha\right)=\begin{pmatrix}I_K & \Delta tI_K\\ [0]_K & \left(I_K+\Delta t\boldsymbol{A}\left(\alpha\right)\right).\end{pmatrix} \tag{3.18}$$

The system 3.17 is unconditionally stable in time if and only if the eigenvalues of $I_K + \Delta t \boldsymbol{A}(\alpha)$ are in the unit disk of the complex plane.

4 Numerical validation on a given configuration

In order to prove the efficiency of the DMD model, we consider the dynamics of a capsule flowing in a microfluidic channel for $\boldsymbol{\theta} = (Ca = 0.1, a/\ell = 0.9)$ and $\Delta t = 0.04$. The DMD method has been applied by considering the evolution of capsule deformability over a non-dimensional time $\dot{\gamma}t \in [0, 10]$. The POD is applied to the displacement snapshot matrix $\mathcal{U}$. We start the validation by performing a singular value decomposition analysis on the displacement snapshot matrix. A determination of the truncation rank is possible by studying the behavior of the relative information content (1−RIC).

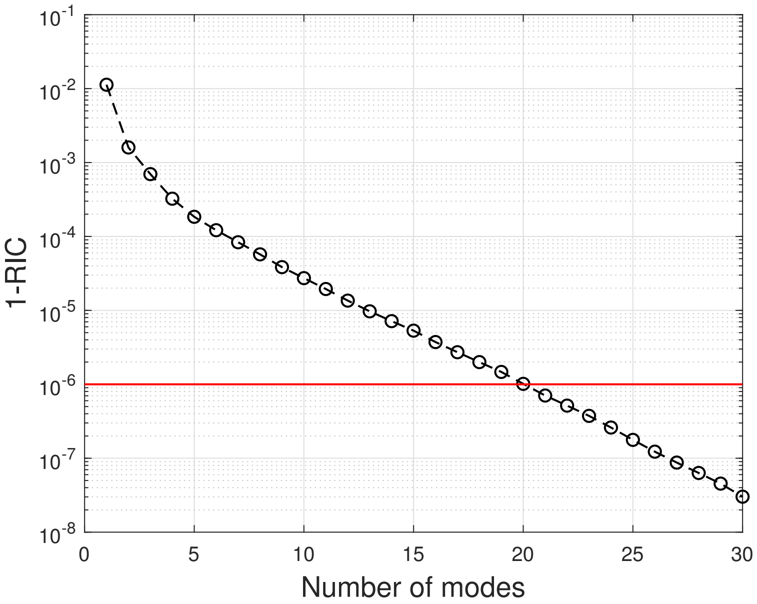


Figure 4.1: The behavior of the relative information as function of truncation rank on the number of modes for the configuration $(Ca = 0.1, a/\ell = 0.9)$.

Figure 4.1 illustrates the evolution of POD of the displacement snapshot matrix according to number of modes. It shows that one mode allows retaining 99% of the total information. We take for our validations 20 modes which correspond to 10^{-6} of accuracy.

4.1 Dynamic mode decomposition resolution

In order to determine the matrix $\boldsymbol{A}$, we construct the matrices $\mathbb{X}$ and $\mathbb{Y}$ as they are expressed in the formulas 3.4. The condition numbers of the matrices $\mathbb{X}$ and $\mathbb{X}\mathbb{X}^T$ give 7.9×10^5 and 6.3×10^9, respectively. These numbers are very high, which may affect the resolution's robustness when it comes to the identification of $\boldsymbol{A}(\alpha)$. In order to check the stability of the model, we plot the eigenvalues of the matrix $\boldsymbol{A}_1(\alpha) = \boldsymbol{A}_0 + \alpha_1 \boldsymbol{A}_1$ for 20 modes.

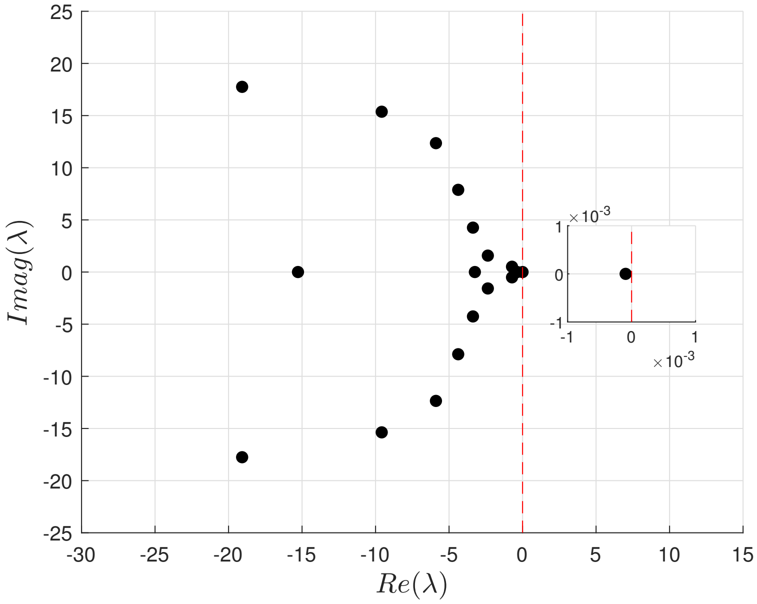


Figure 4.2: Complex eigenvalues $\lambda_k = Re(\lambda_k) + i\mathrm{Imag}(\lambda_k)$, for $k = 1, \ldots, 20$ of the matrix $\boldsymbol{A}_1(\alpha) = \boldsymbol{A}_0 + \alpha_1 \boldsymbol{A}_1$ when $\boldsymbol{\theta} = (Ca = 0.1, a/\ell = 0.9)$ and 20 modes are considered.

Figure 4.2 shows that the complex eigenvalues of $\boldsymbol{A}_1(\alpha)$ have negative real parts which ensure the stability of the system. We also study the stability of the system with Tikhonov regularization 3.8 which is applied according to the regularization coefficient μ. To determine this latter, we plot the residual $\|\mathbb{Y} - \boldsymbol{A}_{1,\mu}(\alpha)\,\mathbb{X}\|_F^2$ as a function of the norm $\|\boldsymbol{A}_{1,\mu}(\alpha)\|_F^2$ where $\boldsymbol{A}_{1,\mu}(\alpha) = \boldsymbol{A}_{0,\mu} + \alpha_1 \boldsymbol{A}_1$.

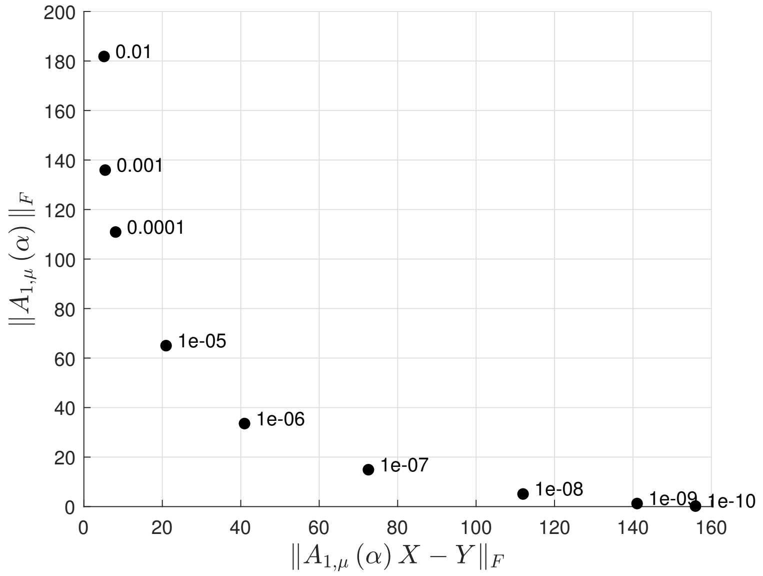


Figure 4.3: Evolution of the residual $\|\boldsymbol{A}_{1,\mu}(\alpha)X - Y\|_F$ as a function of the norm solution $\|\boldsymbol{A}_{1,\mu}(\alpha)\|_F$ when $\boldsymbol{\theta} = (Ca = 0.1, a/\ell = 0.9)$ and 20 modes are considered.

Since the choice of the regularization must be made while guaranteeing the system stability, we plot the eigenvalues of the matrix $\boldsymbol{A}_{1,\mu}(\alpha)$ for a different values of μ and 20 modes.

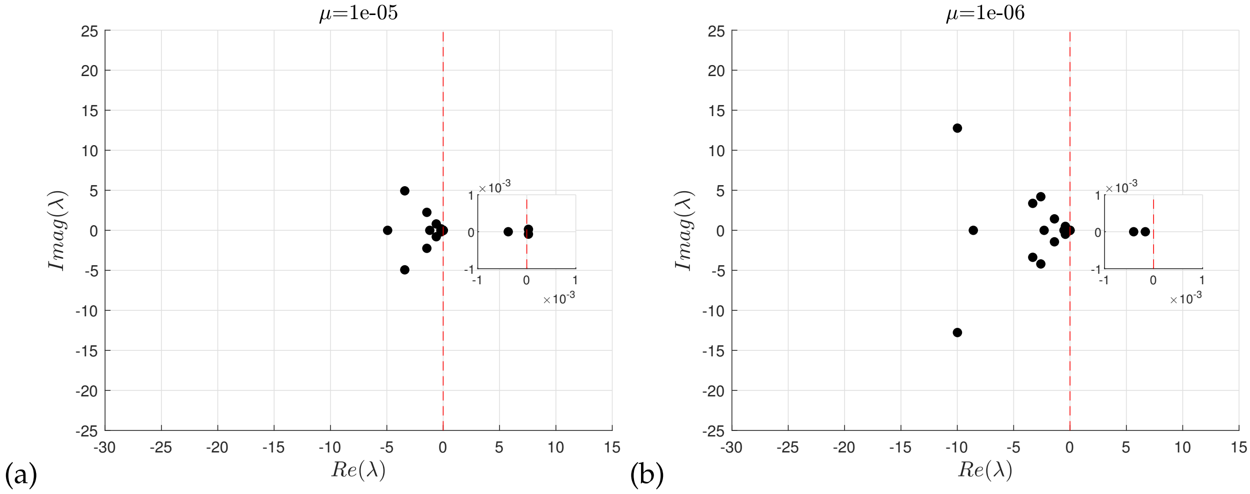


Figure 4.4: Eigenvalues of $\boldsymbol{A}_{1,\mu}(\alpha)$ with $r = 20$ modes for :(a) $\mu = 10^{-5}$; (b) $\mu = 10^{-6}$.

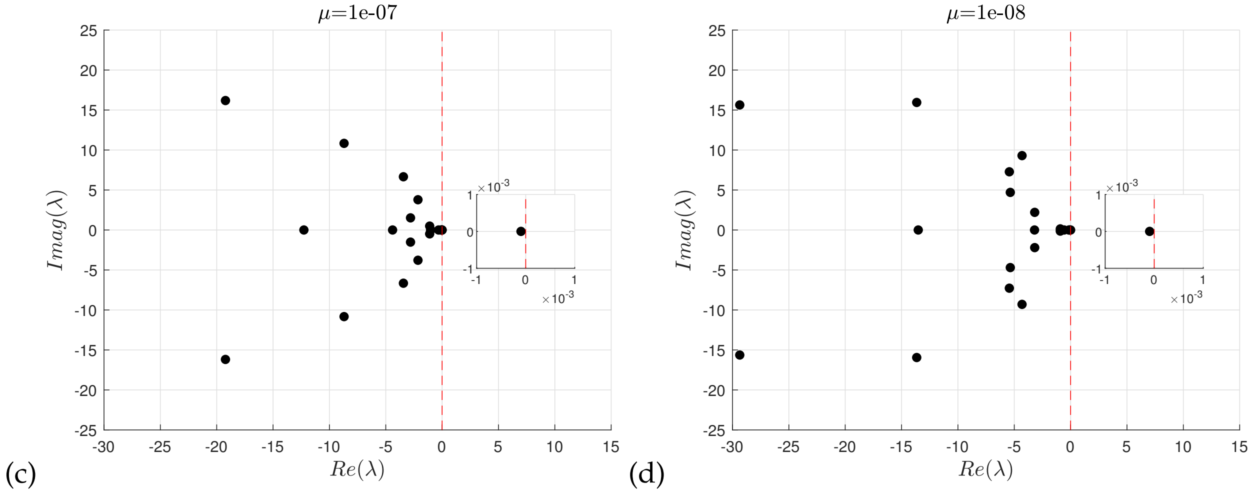


Figure 4.5: Eigenvalues of $A_{1,\mu}(\alpha)$ with $r = 20$ modes for :(c) $\mu = 10^{-5}$; (d) $\mu = 10^{-8}$.

Figures 4.4-4.5 show that the system becomes stable with very small regularization $\mu < 10^{-5}$.

4.2 DMD prediction accuracy

The DMD model allows the prediction of capsule deformability through time. It predicts the state of capsule displacement at t^{n+1} from the previous state at t^n. To check the accuracy of the model, we compare the DMD capsule deformation results with FOM simulations. We perform a prediction of the capsule deformation beyond the considered non-dimensional time $\dot{\gamma}t = 10$ in the database. The prediction test accuracy is done by estimating the capsule deformation for $\dot{\gamma}t = 15$. The test is done with and without regularization. The prediction accuracy over time is measured by computing normalized Modified Hausdorff distance between the obtained predicted capsule shapes and FOM results.

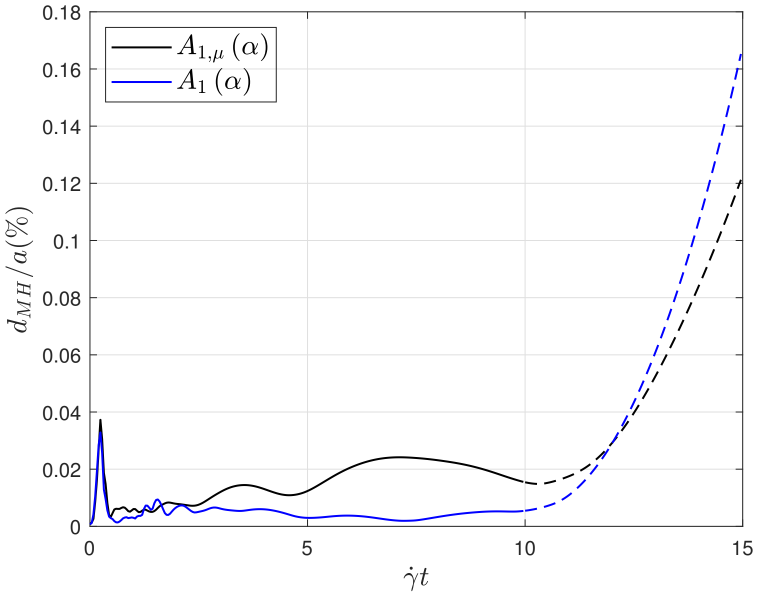


Figure 4.6: Accuracy evolution of the capsule shape prediction for the configuration $\boldsymbol{\theta} = (Ca = 0.1, a/\ell = 0.9)$ using DMD model with 20 modes. (continuous line) prediction in the database time interval; (dashed-line) prediction out of the database time interval. (black-red line) $\boldsymbol{A}(\alpha) = A_0 + \alpha \boldsymbol{A}_1$ computed without regularization formula 3.8. (blue line) $\boldsymbol{A}_\mu(\alpha) = A_\mu + \alpha \boldsymbol{A}_1$ computed using regularization formula where $\mu = 10^{-8}$ and $\dot{\gamma} = V/\ell$.

The figure 4.6 illustrates the prediction accuracy against FOM solutions. As shown, the Tikhonov regularization gives a less accurate prediction than the case without regularization at the database interval. However, beyond the database time interval $\dot{\gamma}t = 10$, the regularization allows having relatively more accurate prediction beyond the database time interval. This differences are due to the taken trade-off between the residual $\|\mathbb{Y} - \boldsymbol{A}(\alpha)\mathbb{X}\|_F^2$ and the norm solution $\|\boldsymbol{A}(\alpha)\|_F^2$.

We investigate the effect of $\boldsymbol{A}(\alpha)$ order on the DMD model efficiency. In figure 4.7, we measure the prediction accuracy through time without regularization for 3 cases: $\boldsymbol{A}_0(\alpha)$, $\boldsymbol{A}_1(\alpha)$, and $\boldsymbol{A}_2(\alpha)$.

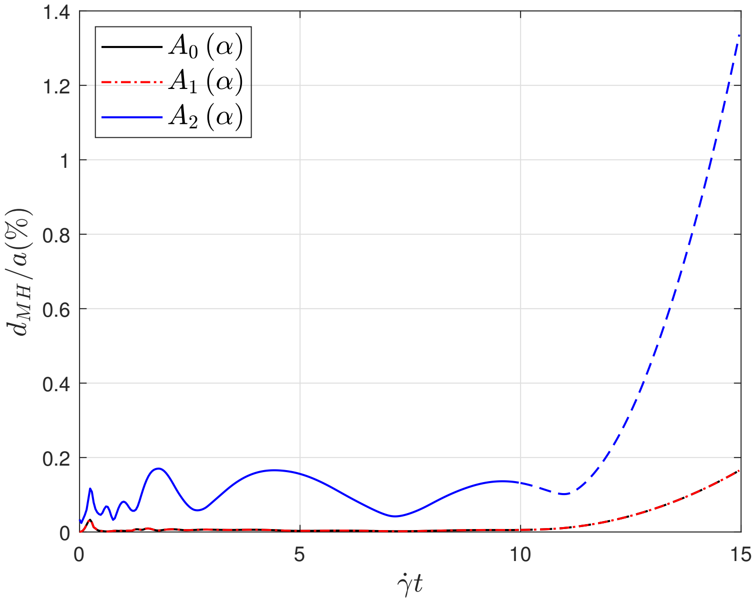


Figure 4.7: Accuracy evolution of the capsule shape prediction for the configuration $\boldsymbol{\theta} = (Ca = 0.1, a/\ell = 0.9)$ using DMD model with 20 modes using $\boldsymbol{A}_0\,(\alpha) = \boldsymbol{A}_0$ (black line), $\boldsymbol{A}_1\,(\alpha) = \boldsymbol{A}_0 + \alpha_1\boldsymbol{A}_1$ (red dash-line), and $\boldsymbol{A}_2\,(\alpha) = \boldsymbol{A}_0 + \alpha_1\boldsymbol{A}_1 + \alpha_2\boldsymbol{A}_2$ (blue line).

As it can be noticed, the accuracy of the model remains the same when $\boldsymbol{A}_0\,(\alpha)$ and $\boldsymbol{A}_1\,(\alpha)$ are considered. However, the prediction becomes less accurate for the second iteration $\boldsymbol{A}_2\,(\alpha)$. The accuracy of prediction is decreased to 1.5%. Beyond the second iteration $\boldsymbol{A}_n\,(\alpha)$, for $n > 2$, the model fails to predict the capsule shape evolution totally. Similar remarks are taken when Tikhonov regularization is considered. The model was unable to perform prediction for $n \geq 1$.

The comparison is done for the configuration $\theta = (Ca = 0.1, a/\ell = 0.9)$ at different states $\dot{\gamma}t = 0, 1, 2, 8$. Figure 4.8 visualize the excellent prediction of the capsule deformation using the DMD approach compared to reference capsule deformation with FOM.

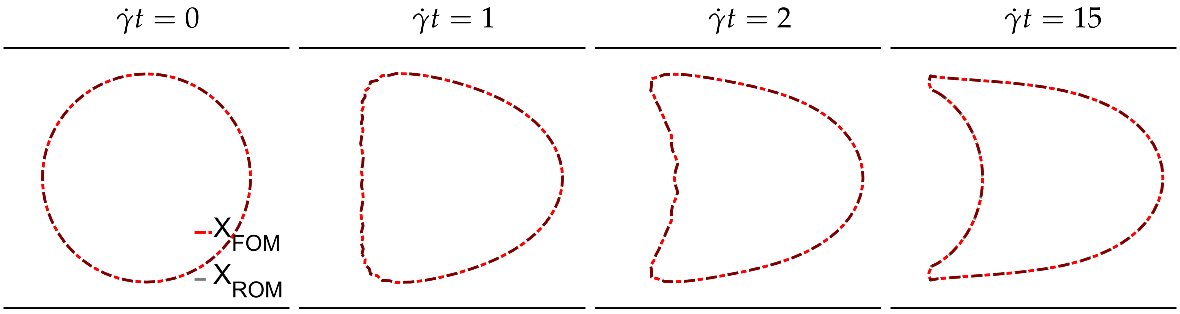


Figure 4.8: Comparison of the capsule profile between the FOM (black dashed line) and the DMD predicted (red dashed line) solutions for $\boldsymbol{\theta} = (Ca = 0.1, a/\ell = 0.9)$ at times $\dot{\gamma}t = 0, 1, 2$ and 15 respectively. The horizontal lines correspond to the channel walls.

5 Concluding remarks

In this chapter, we presented a dynamic mode decomposition model (DMD-ROM) to simulate capsule deformability in a microfluidic channel. It involves the construction of a reduced-order dynamic system using a POD procedure. The system is constructed with a non-constant matrix $\boldsymbol{A}(\alpha)$ that expresses the dynamic of capsule deformation in time. To identify the matrix $\boldsymbol{A}(\alpha)$ a dynamic mode decomposition method is used. The nonconstant matrix $\boldsymbol{A}(\alpha)$ is computed under the form of $\boldsymbol{A}_k(\alpha) = \boldsymbol{A}_0 + \sum_{k\geq 1}^{K} \alpha_k \boldsymbol{A}_k$. It has been shown that the eigenvalues of $\boldsymbol{A}(\alpha)$ have negative real parts, which ensure the stability of the dynamic system. For stability reasons $\boldsymbol{A}(\alpha)$ is constructed at just one iteration. Any extended form has led to divergent results. The numerical validation of the model gives clear evidence of its reliability in predicting the capsule deformation of any desired state with an acceptable error rate.

Chapter 6

Conclusions et perspectives

Throughout this thesis dissertation, we aimed to propose alternative fast-simulation models to the high-fidelity model used to solve the problem of deformable capsules flowing in microfluidic channels. To seek this objective, we were interested in exploring data-driven approaches based on ROM+manifold learning and ROM+DMD techniques. The ROM+manifold learning approach is based on the construction of a low-order space-time-parameter basis model. This latter allows the prediction of any query admissible couple $\boldsymbol{\theta}_q = (Ca\,, a/\ell)$ in the parameter domain. A manifold learning is then achieved by the use of time-snapshot data of the interpolated solution at $\boldsymbol{\theta} = \boldsymbol{\theta}_q$. One of the most outstanding features of this approach is fast-simulation computations that can be done in few seconds, which sharply contrasts with the days required by the FOM method.

The Data-driven ROM+DMD approach is based on constructing a ROM dynamic system of the treated problem. The dynamic system describes the dynamic relationship between the spectral coefficients of the POD decompositions for the displacements and velocity fields. The key idea is the identification of a dynamic operator that expresses the dynamic relation, where we suggested using a dynamic mode decomposition approach.

As far as we know, the first model is the first dynamic reduced-order models for treating the time-evolution of capsule deformability. Numerical validations confirm the efficiency of the two methods where the simulation speedups are observed to be of order 10,000 with less than 0.3% in error average for ROM+manifold learning and less than 0.2% as a maximum error for ROM+DMD method.

We also developed a relevant software tool with an intuitive interface to explore the time evolution of capsule deformation for any desired admissible values of capillary number and confinement ratio. This software tool uses the data-driven reduced model technique to prediction the three dimensional deformation of capsule flowing in microfluidic channel very quickly.

Furthermore, we believe that the proposed approaches in this thesis can be applied to a broad range of multiphysics problems such as fluid-structure interactions, structural dynamics using quasi-static structural mechanics models, and related problems.

As far as possible future work perspectives are concerned, several interesting research avenues have been identified throughout the present thesis work. We aim to use the fast-simulation tools to identify the mechanical properties from experimental results of capsule deformation. The properties will be determined by fitting the deformed capsule shape predicted by the reduced-order models to the experimental one using diffuse approximation techniques. The case of more complex FSI configurations, such as the deformation of capsules flowing through a Y-shaped microchannel bifurcation, will be investigated in future work. For the dynamic data-driven model, the method is not fully optimized to be more efficient. Nevertheless, improvements should be brought to the approach before going to the next level. For instance, we plan to construct a dynamic global model enabling the prediction of any parameter vector from the admissible parametric space. Finally, we are also interested in using artificial neural networks (ANN) as an alternative to the DMD approach to deal with a nonlinear algebraic system.

Appendix A

Inverse analysis of capsule mechanical properties flowing in a microfluidic channel

Contents

1 Introduction

The behavior of a capsule flowing through a microfluidic channel lies mainly on the hydrodynamics of the flow that it is subjected to and its membrane mechanical properties. In this chapter, we post-process the obtained data set from FOM to investigate the link between the measurement of characteristic lengths on the capsule deformed profiles and their capillary number and confinement ratio. The data set of FOM resolutions forms an admissible domain for space-time-parameters solutions. We will also try to gather similar cases into clusters that will be helpful in performing the ROM model in the next chapter.

2 Database of FOM results

2.1 Membrane characteristics of numerical results

In this section, we study the numerical results of FOM to determine the mechanical properties of capsules flowing in a square-section microfluidic channel.

We take the measurements presented in figure 2.1 from the capsule profile.

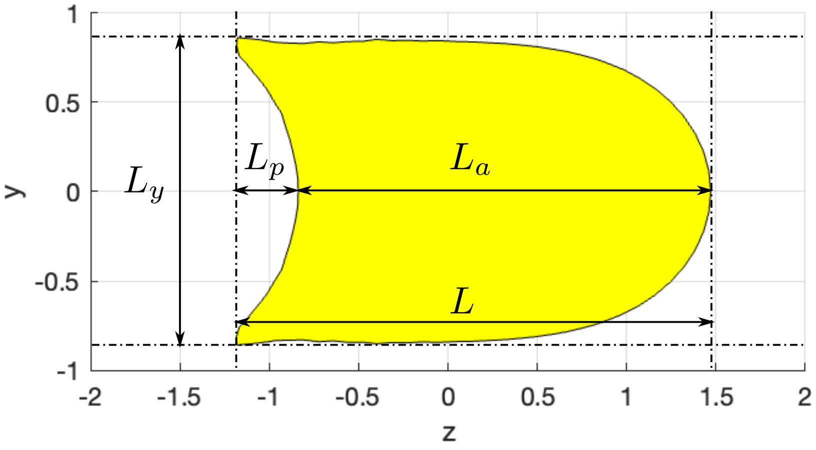


Figure 2.1: The capsule membrane measurement dimensions in profile presentation.

- S/ℓ^2: the surface of the profile.
- L/ℓ: the total length.
- L_a/ℓ: the axial length.
- L_p/ℓ: the parachute depth which is given by $L_p = L - L_a$.

The results of the numerical model are gathered in charts, where the main output parameters, i.e., total length L/ℓ, parachute depth L_p/ℓ are plotted as functions of Ca and a/ℓ for capsules with a NH membrane.

We note that the capsule elongation is proportional to the capillary number and aspect ratio. Meanwhile, the capsules width decreases relatively for those with a confinement ratio under $a/\ell < 0.95$, which becomes much important otherwise.

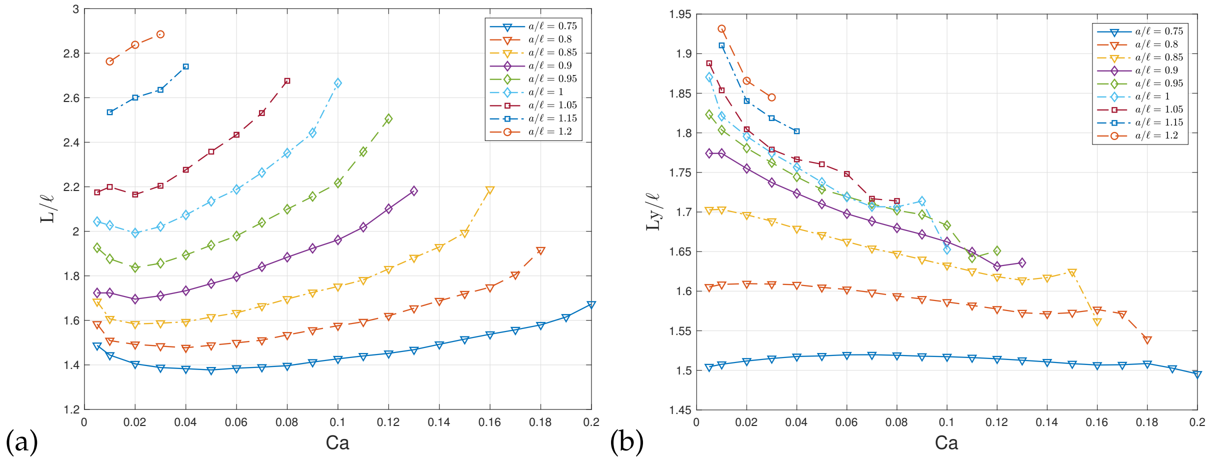


Figure 2.2: Capsule total length L/ℓ, (b) axial width L_y/ℓ at steady-state flowing in a square-channel.

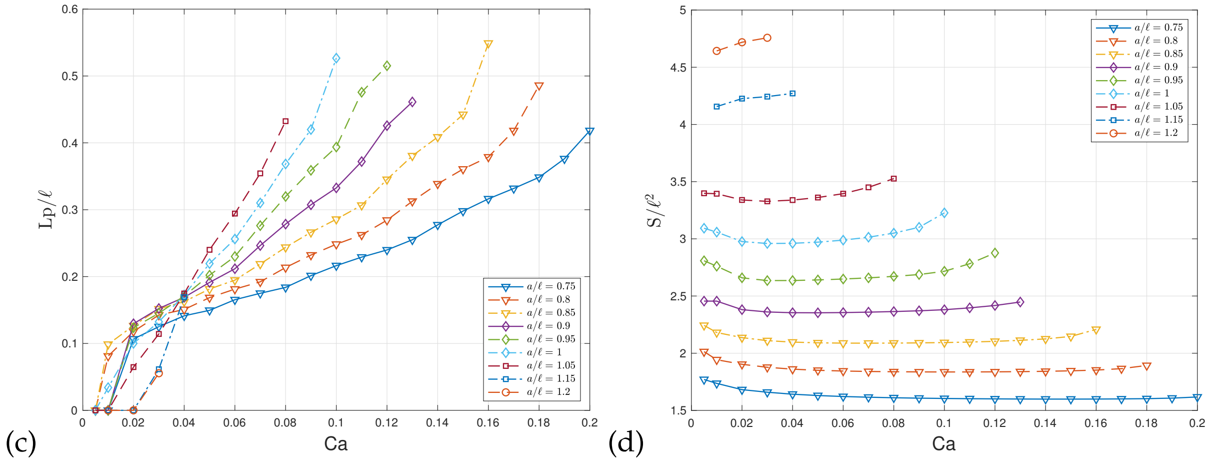


Figure 2.3: (c) parachute depth L_p/ℓ; (d) surface of capsule profile S/ℓ^2.

Figure 2.3 shows that the parachute depth depends mainly on the capillary number, where the higher the capillary number, the more the parachute shape becomes more shaped. The surface depends on the initial size of the capsule. The relative decrease of the surface size in most cases is related to the deformation of the membrane surface, and the increase of the same quantities for some cases is done due to the elongation phenomena it started to appear at high capillary number values. Furthermore, the presented numerical results in figures 2.2-2.3 can be used to determine physical proprieties as the shear modulus G_s through an inverse analysis (Hu *et al.*, 2013; Sévénié *et al.*, 2015), or an artificial neural network approach.

3 Clustering of parametric space

Clustering or unsupervised classification is a technique to group similarities in separable clusters (Fraley & Raftery, 1998). Partitioning Clustering aims to partition n objects into k clusters based on a measure of similarity such that similarities between data in the same group are high while similarities between data in different groups are low. Thus, clustering parametric space allows maximizing the efficiency of the reduced-order model since it will be applied to similar cases.

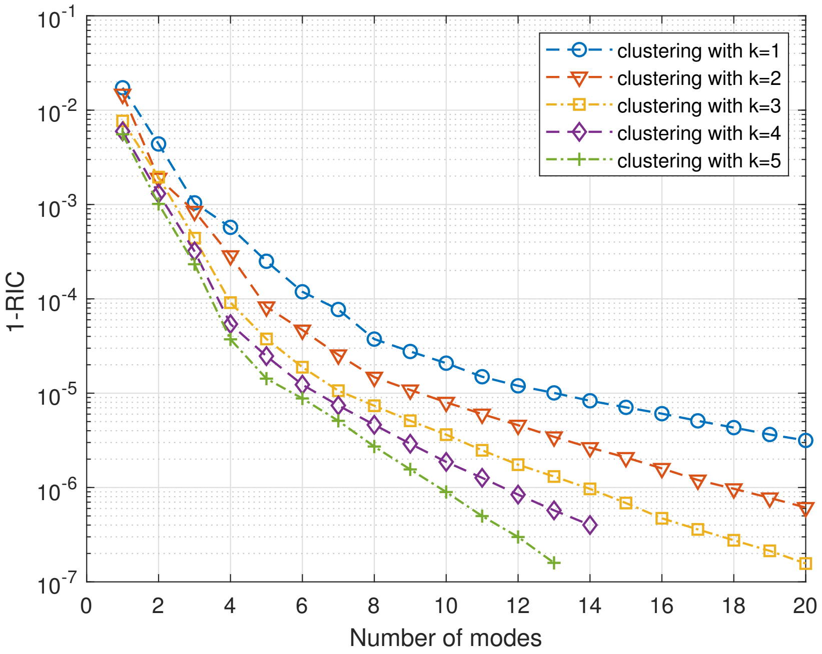


Figure 3.1: Spectrum singular values behavior according to the number of clusters. The clustering is applied using the K-means algorithm, and capsules positions fields at the steady-state $\dot{\gamma}t = 20$.

Figure 3.1 shows that spectrum of singular values decreases more quickly by clustering rather than taking all the parametric space. In addition, clustering can reduce computation costs of constructing the data-driven reduced bases, as will be detailed in the next chapter.

Many clustering algorithms of partitioning are done by creating disjoint clusters, which means that each object is assigned to one cluster. For instance, K-means is one of the most popular partitioning clustering algorithms (Jian, 2009). However, it would be recommended for many clustering applications to tolerate overlaps between clusters so that an object may belong to one or more clusters. Overlapping clustering has been studied through various approaches during the last decades, which can be divided into two main categories: hard memberships and uncertain memberships (Cleuziou, 2007; Afridi *et al.*, 2020). Hard memberships or k-means-like approaches are one of the widely used approaches. They are based on Euclidean distance function, and their particularity resides in their fast and low computational complexity, making them suitable for mining large data sets. Uncertain memberships can be denoted as the solutions which model clusters memberships for each data object as uncertainty function using fuzzy or possibilistic (Höppner *et al.*, 1999; Hruschka *et al.*, 2009). In the following sections, we use, for simplicity reasons, k-means variants to clustering the parametric space 2.1 at the steady-state.

3.1 K-means clustering

K-means proposed 50 years ago (Ball & Hall, 1965), it aims to find k centroids, such that one element for each cluster. It is done by minimizing the sum of the distance of each datum from its respective cluster centroid. In other words, for $x_i \in X$, we can write:

$$\underset{\{C_1,\ldots,C_k\}}{argmin} \sum_{j=1}^{k} \sum_{x_i \in C_j} d\left(x_i, c_j\right) \tag{3.1}$$

where $(C_1, \ldots, C_k)$ are k non-overlapping clusters, c_j is the representative of cluster C_j, and d is a distance function.

3.2 Identification of clustering criteria

A suitable clustering depends on the used considered geometrical propriety ($\{\boldsymbol{x}\}$, L_p, L_a, etc). In this section, we initially perform a k-means clustering for a fixed number of clusters ($k = 3$) and different measures. This allows to investigate the best measure to use for clustering the parametric space.

At the steady-state, visually one can distinguish 3 main categories of the capsules shape in the parametric space:

- capsules with less deformation are the ones who have a small capillary number.
- capsules with a parachute shape are the ones with significant capillary number.
- capsules with an elongated shape are located above $a/\ell \geq 0.95$.

From these observations, one can determine the most relevant clustering in figures 3.2-3.3 are the clustering with the positions of membrane nodes at the steady-state. In the next section, we study the use of overlapping clustering with NEO-k-means approach.

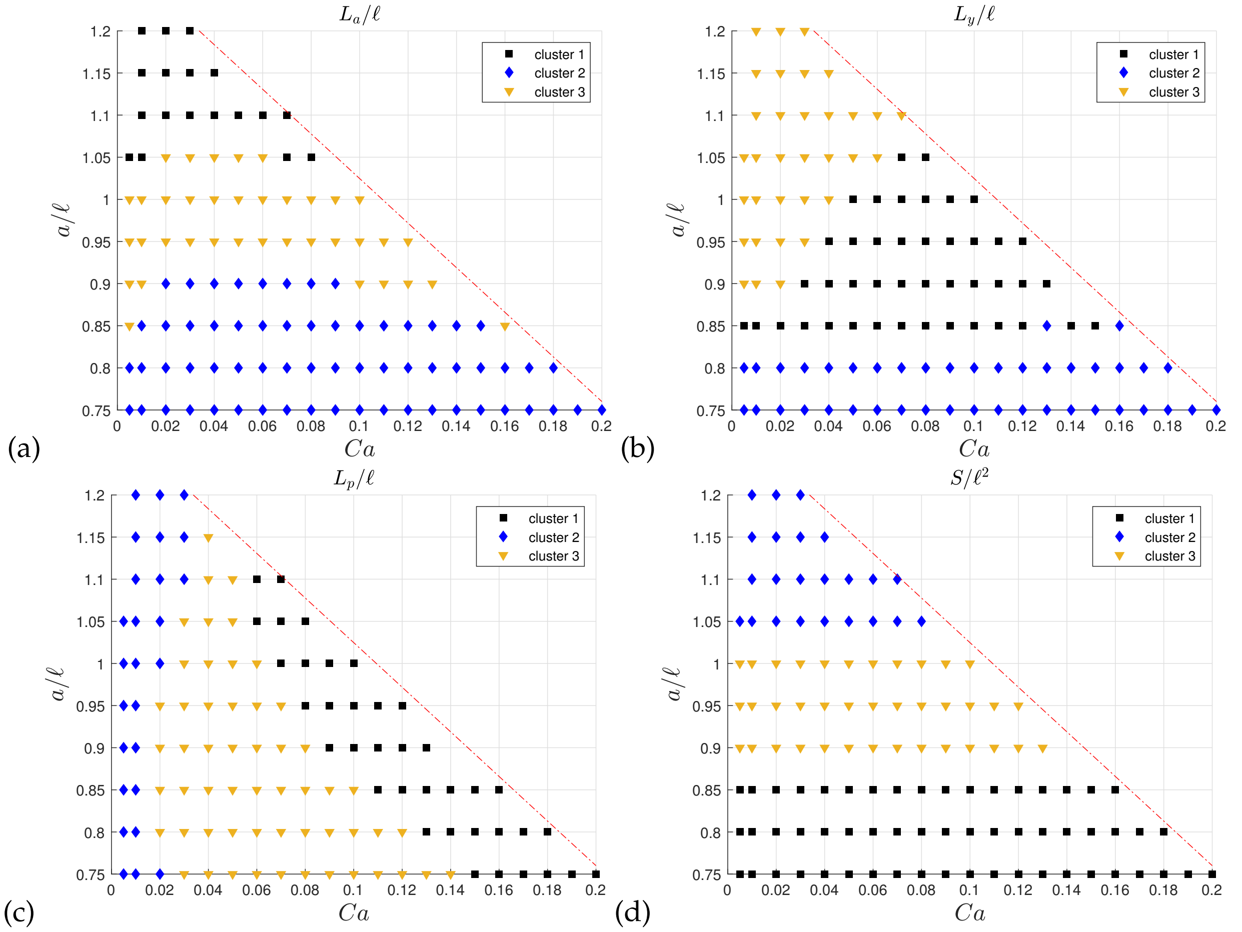


Figure 3.2: Clustering of parametric space to 3 clusters according to (a) *La* the axial length (b) *Ly* the axial width; (d) (S) the surface of the profile in the axial plane.

3.3 Overlapping clustering

The use of an overlapping clustering approach is highly recommended for our needs since the prediction of parameter vector at the border of a domain may leads to less accurate results. For that we use Non-Exhaustive Overlapping k-means clustering (Whang *et al.*, 2019) which is approach based on the introduction of an assignment matrix $U = [u_{ij}]_{n\times k}$ such that

$$u_{ij} = \begin{cases} 1 & \text{if } x_i \in j\text{th cluster} \\ 0 & \text{otherwise.} \end{cases} \tag{3.2}$$

this means that there is no restrictions on the assignment matrix U, in such way a data point can belong to multiple clusters. For that, it is mandatory to fix the number of assignments. To control the number of additional assignments, a constraint will be added to the total of assignments in U should be equal to $n + \alpha n$. The application of this constraint alone can lead

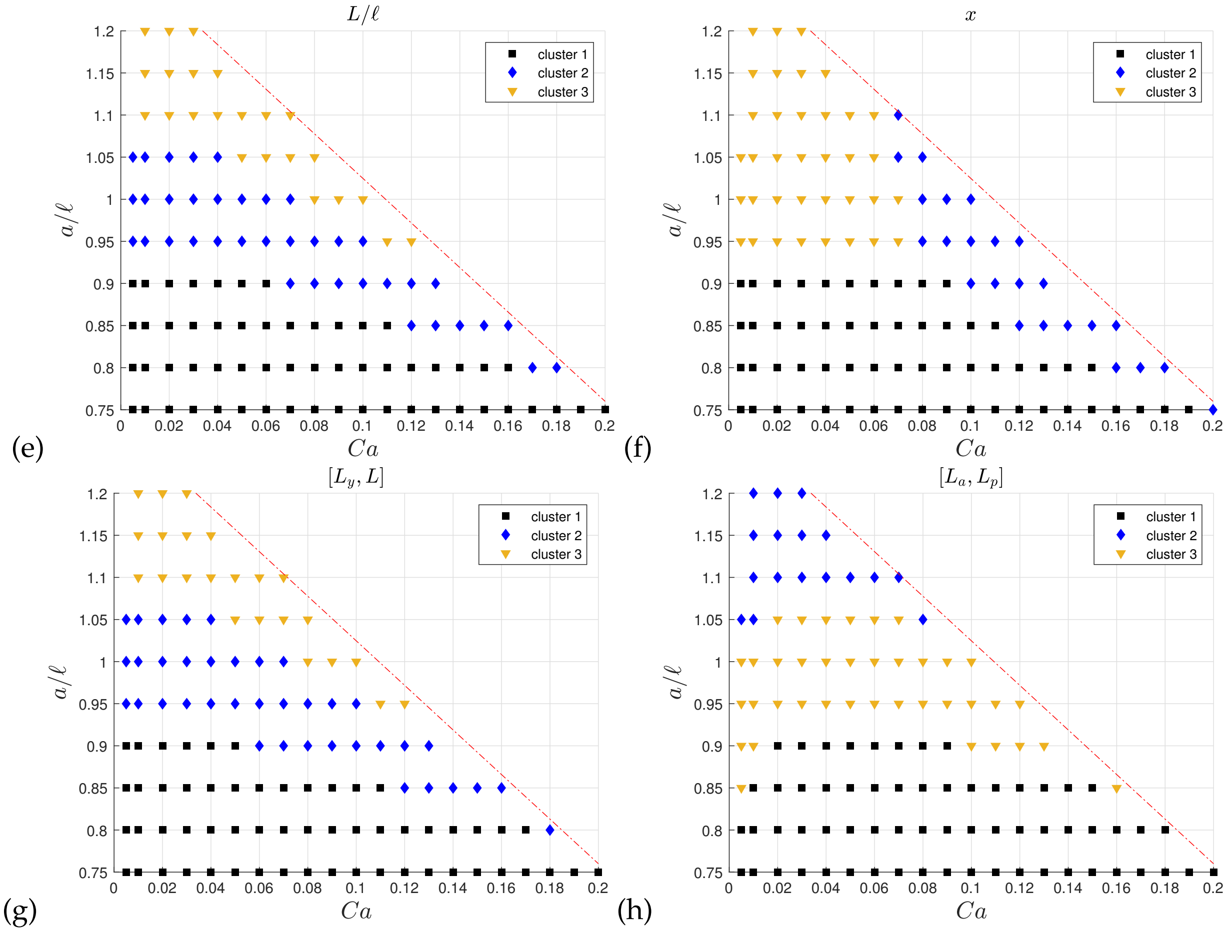


Figure 3.3: Clustering of parametric space to 3 clusters according to (e) L total length; (f) x position of the membrane nodes; (g) $[Ly, L]$ axial width with the total length (h) $[La, Lp]$ axial length with the parachute depth.

to false outliers. To avoid that another constraint will be introduced to control the degree of non-exhaustiveness. So, the NEO-k-Means objective function can be given as:

$$\min_{U} \sum_{j=1}^{k} \sum_{i=1}^{n} u_{ij} \|x_i - m_j\|^2, \text{ where } m_j = \frac{\sum_{i=1}^{n} u_{ij} x_i}{\sum_{i=1}^{n} u_{ij}} \tag{3.3}$$

such that trace $\left(U^T U\right) = (1+\alpha)\, n$, $\sum_{i=1}^{n} \mathbb{I}\{(U\mathbf{1})_i = 1\} \leq 0$, where

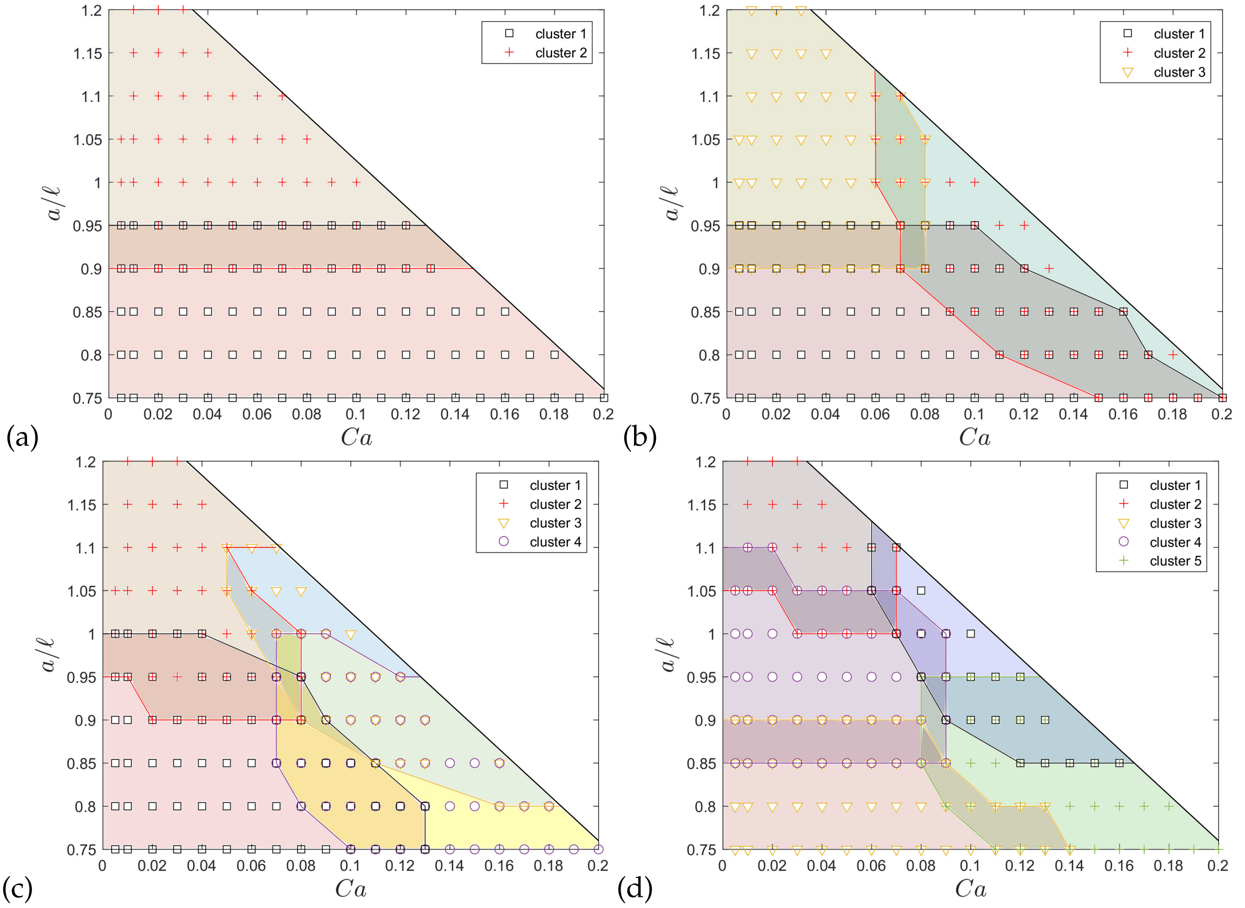
$$\mathbb{I}\,(expression) = \begin{cases} 1 & \text{if expression is true} \\ 0 & \text{otherwise.} \end{cases} \tag{3.4}$$

and the vector $U\mathbf{1}$ denotes the number of clusters to which each data point belongs where $m1$ denotes a $k \times 1$ column vector having all the elements equal to one. Thus, $(Um1)_i = 0$ means that

x_i does not belong to any cluster. The parameter α allows capturing the degree of overlap and non-exhaustiveness. If $\alpha = 0$ the NEO-K-Means objective function is equivalent to the standard k-means.

The application of Neo-K-means according to the positions $/bx$ at the steady-state for different number of clusters

We performed neo-kmeans on capsule positions at the steady-state ($\dot{\gamma}t = 20$) of parametric space with $\alpha = 2.5$.



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

Figure 3.4: Partitioning the parametric domain for a different number of clusters using NEO-k-means according to membrane node positions at the steady-state. (a) 2 clusters; (b) 3 clusters; (c) 4 clusters; (d) 5 clusters.

As figure 3.4 shows, the more the number of clusters increases, the more the clustering is determined through more details of the capsule shapes. For instance, the partitioning with two clusters gives two subdomains divided at the pre-deformed zone limit $a/\ell = 0.95$, figure 3.4(a). For 3 or 4 clusters, in addition to the previous partitioning, the new cluster compromises parachute shapes at the right of the domain 3.4(b-c). For 5 clusters, a new cluster takes the left middle of the domain figure 3.4(d).

4 Conclusion

In this chapter, we have studied the characteristic dimensions of obtained FOM simulation results. This latter constitutes a parametric domain of admissible solutions that represents a crucial element in the construction of a ROM approach. The clustering of the parametric domain is a relevant task since it can maximize the new model's performance. There are different clustering methods; for our need, we choose non-Exhaustive Overlapping K-means clustering since it ensures a partitioning of the parametric domain with overlapping sections. Furthermore, this will ensure a good inspection of the cases located at the border of the clusters. The deployment of the clustering procedure needs mainly to determine two parameters: the clustering criteria and the number of clusters. The investigation of clustering with different parameters shows that the position field gives the partitioning with the best description of the parametric domain.

Appendix B

Optimal Experimental Design (OED)

Despite the outperforming results obtained with the data-driven reduced-order model compared to the full order model, some adjustments are needed to allow the software tool presented above to be the fastest possible. One of them is the optimization of the parametric design (OPD) instead of the full experimental design (FED). The OPD allows decreasing the computation cost much more. To seek this goal, we suggested using experimental design (OED) generation techniques as Latin Hypercube design. This latter allows constructing an OED with a minimum number of configurations while maintaining accurate predictions.

1 Construction of an Optimal Experimental Design (OED)

The construction of OED is done through the code that uses the native MATLAB function lhs. It takes as inputs the number of generated configurations n, its dimensions p, the minimization that is used to compute the sum between-column squared correlations, and the number of iterations to improve the criterion.

It returns a Latin hypercube matrix X of size $n \times p$. Each column of X contains n randomly distributed values with one from each interval $(0, 1/n), (1/n, 2/n), \ldots, (1 - 1/n, 1)$, and randomly permuted.

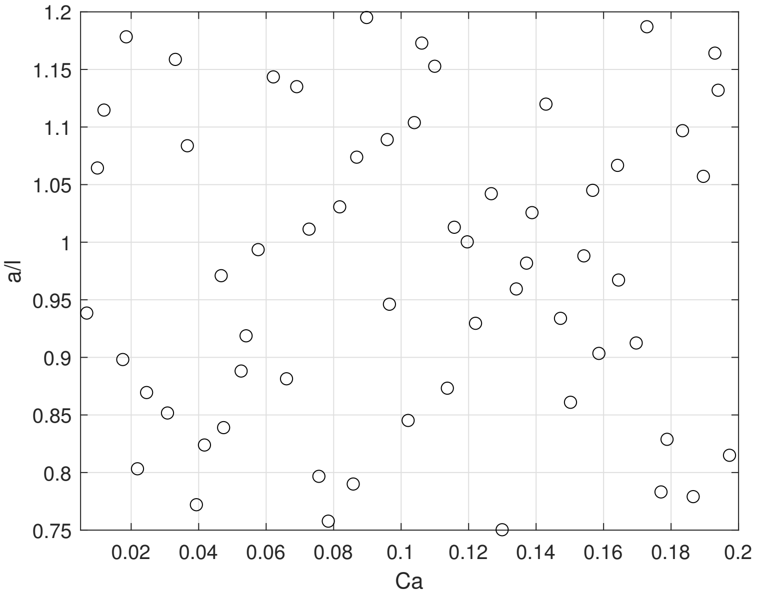


Figure 1.1: Experimental design with Latin Hypercube design.

By limitation of the region of interest, the code gives a first optimal experimental design (see figure 1.2).

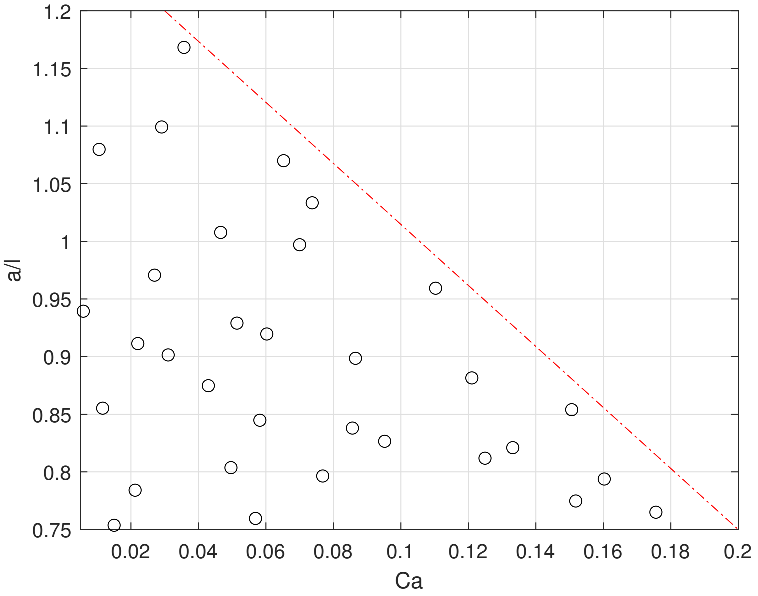


Figure 1.2: Experimental design with 32 generated configurations.

The generation of OED can be repeated if it is needed. Once the initial OED is fixed, the code proceeds to fit it to the existing database of 118 configurations using a nearest to the nearest procedure:

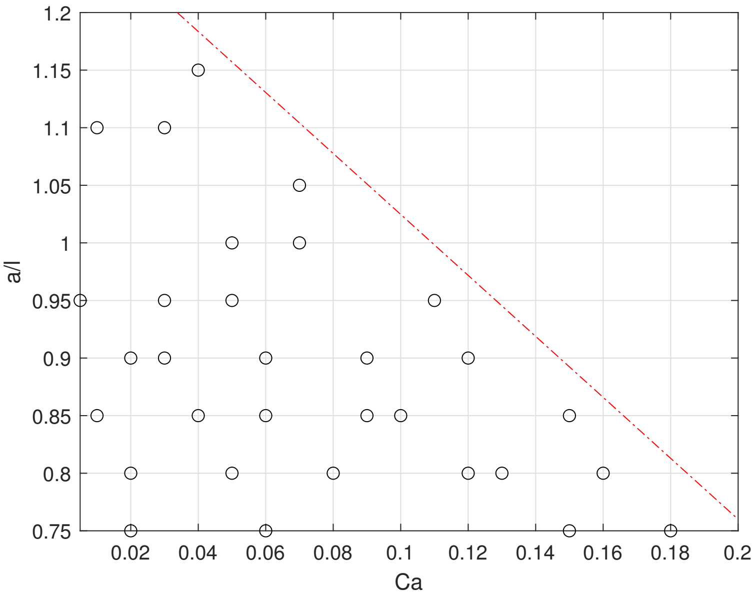


Figure 1.3: Latin Hypercube design is fitted to the existing database.

The code allows then the user to enrich the obtained OED or exclude not desired configurations.

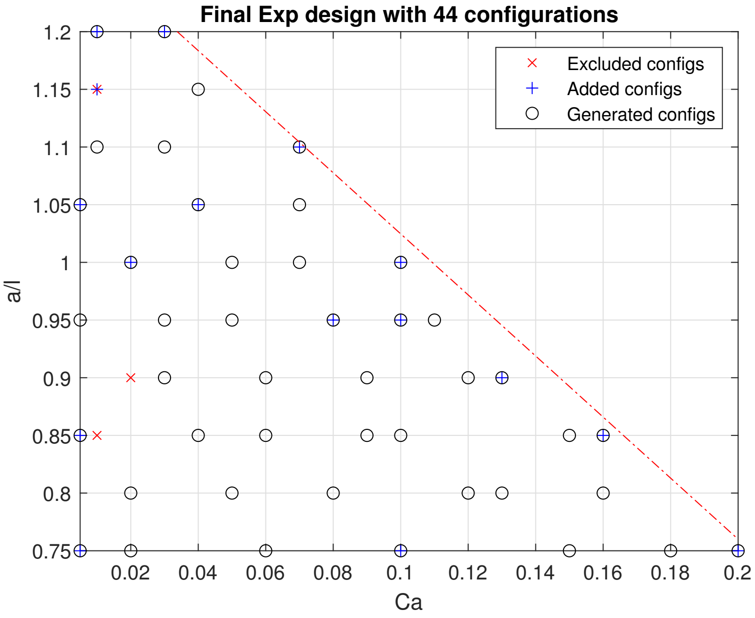


Figure 1.4: Modified Latin Hypercube design. ◦ : generated configuration; × : excluded configurations; + : new configurations.

As a result, user get the final OED presented in figure-(1.5) in mat-file form.

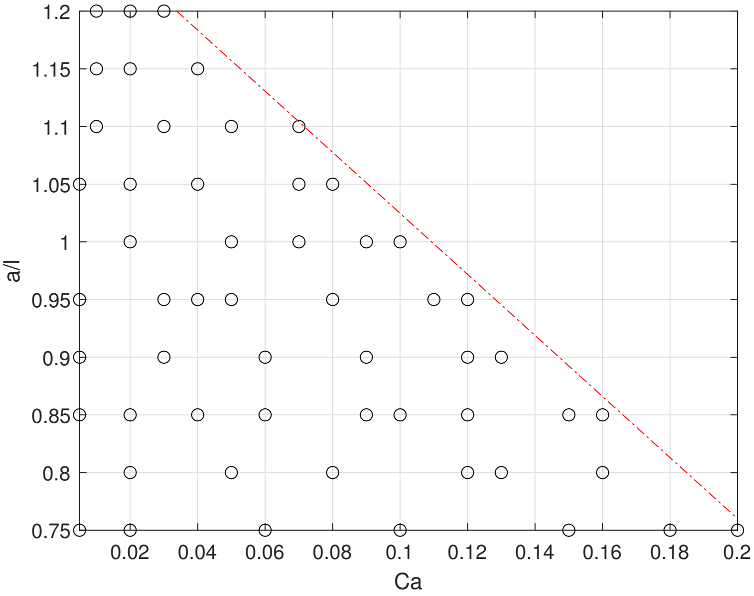


Figure 1.5: Optimal experimental design.

Bibliography

ABREU, M. S. C., MORENO, M. J. & VAZ, W. L. C. 2004 Kinetics and thermodynamics of association of a phospholipid derivative with lipid bilayers in liquid-disordered and liquid-ordered phases. *Biophysical Journal* **87** (1), 353–365.

AFRIDI, M. K., AZAM, N. & YAO, J. 2020 Variance based three-way clustering approaches for handling overlapping clustering. *International Journal of Approximate Reasoning* **118**, 47 – 63.

AUBRY, N. 1991 On the hidden beauty of the proper orthogonal decomposition. *Theoretical and Computational Fluid Dynamics* **2** (5), 339–352

AUBRY, N., HOLMES, P., LUMLEY, J. L. & STONE, E. 1988 The dynamics of coherent structures in the wall region of a turbulent boundary layer. *Journal of Fluid Mechanics* **192**, 115–173.

BALL, G. H. & HALL, D. J. 1965 Isodata, a novel method of data analysis and pattern classification. *Tech. Rep.*.

BARTHÈS-BIESEL, D. 2011 Modeling the motion of capsules in flow. *Curr. Opin. Colloid In.* **16**, 3 – 12.

BARTHÈS-BIESEL, D., DIAZ, A. & DHENIN, E. 2002 Effect of constitutive laws for two dimensional membranes on flow-induced capsule deformation **460**, 211–222.

BELOV, Y. Y. 2012 *Inverse problems for partial differential equations*. De Gruyter.

BENJAMIN, P. & KAREN, W. 2015 Dynamic data-driven reduced-order models. *Computer Methods in Applied Mechanics and Engineering* **291**, 21–41.

BENJAMIN, P. & KAREN, W. 2016 Data-driven operator inference for nonintrusive projection-based model reduction. *Computer Methods in Applied Mechanics and Engineering* **306**, 196–215.

Benner, P., Gugercin, S. & Willcox, K. 2015 A Survey of Projection-Based Model Reduction Methods for Parametric Dynamical Systems. *SIAM Review* **57** (4), 483–531.

Boubehziz, T., Quesada-Granja, C., Dupont, C., Villon, P., De Vuyst, F. & Salsac, A.-V. 2021 A data-driven space-time-parameter reduced-order model with manifold learning for coupled problems: Application to deformable capsules flowing in microchannels. *Entropy* **23** (9), 1193.

Breitkopf, P., Naceur, H., Rassineux, A. & Villon, P. 2005 Moving least squares response surface approximation: formulation and metal forming applications. *Computers & Structures* **83** (17-18), 1411–1428 0045–7949.

Breitkopf, P., Rassineux, A. & Villon, P. 2002 An introduction to moving least squares meshfree methods. *Revue Européenne des Eléments Finis* **11** (7), 825–867.

del Burgo, L. S., Compte, M., Aceves, M., Hernández, R. M., Sanz, L., Álvarez-Vallina, L. & Pedraz, J. L. 2015 Microencapsulation of therapeutic bispecific antibodies producing cells: immunotherapeutic organoids for cancer management. *Journal of Drug Targeting* **23** (2), 170–179, pMID: 25338126.

Carin, M., Barthès-Biesel, D., Edwards-Lévy, F., Postel, C. & Andrei, D. 2003 Compression of biocompatible liquid-filled HSA-alginate capsules: determination of the membrane mechanical properties. *Biotechnol. Bioeng.* **82**, 207–212.

Cavoretto, R. & De Rossi, A. 2016 Kernel-based methods and function approximation. *Dolomites Research Notes on Approximation* **9** (Special Issue), 2035–6803.

Cerda, E. & Mahadevan, L. 2003 Geometry and physics of wrinkling. *Phys. Rev. Lett.* **90** (7), 074302.

Chang, K.-S. & Olbricht, W. L. 1993 Experimental studies of the deformation of a synthetic capsule in extensional flow. *Journal of Fluid Mechanics* **250**, 587–608

Chaturantabut, S. & Sorensen, D. C. 2010 Nonlinear model reduction via discrete empirical interpolation. *SIAM Journal on Scientific Computing* **32** (5), 2737–2764.

Chen, K. K., Tu, J. H. & Rowley, C. W. 2012 Variants of dynamic mode decomposition: Boundary condition, koopman, and fourier analyses **22** (6), 887–915.

Chen, S. & Doolen, G. D. 1998 Lattice boltzmann method for fluid flows. *Annual review of fluid mechanics* **30** (1), 329–364 0066–4189.

Chinesta, F., Ammar, A. & Cueto, E. 2010 On the use of Proper Generalized Decompositions for solving the multidimensional chemical master equation. *European Journal of Computational Mechanics/Revue Européenne de Mécanique Numérique* **19** (1-3), 53–64 1779–7179.

Chinesta, F., Ladeveze, P. & Cueto, E. 2011 A short review on model order reduction based on Proper Generalized Decomposition. *Archives of Computational Methods in Engineering* **18** (4), 395, ISBN: 1886-1784.

Christensen, E. A., Brøns, M. & Sørensen, J. N. 1999 Evaluation of Proper Orthogonal Decomposition–based decomposition techniques applied to parameter-dependent nonturbulent flows. *SIAM Journal on Scientific Computing* **21** (4), 1419–1434

Chu, T. X., Salsac, A.-V., Leclerc, E., Barthès-Biesel, D., Wurtz, H. & Edwards-Lévy, F. 2011 Comparison between measurements of elasticity and free amino group content of ovalbumin microcapsule membranes: Discrimination of the cross–linking degree. *J. Colloid Interf. Sci.* **355** (1), 81 – 88.

Cleuziou, G. 2007 A generalization of k-means for overlapping clustering. *Rapport technique* **54**.

Cordier, L. 2008 Proper Orthogonal Decomposition: an overview. Lecture series 2008-01 on post-processing of experimental and numerical data, Von Karman Institute for Fluid Dynamics.

Cottet, G.-H., Maitre, E. & Milcent, T. 2008 Eulerian formulation and level set models for incompressible fluid-structure interaction. *ESAIM: Mathematical Modelling and Numerical Analysis - Modélisation Mathématique et Analyse Numérique* **42** (3), 471–492.

Cressie, N. 1988 Spatial prediction and ordinary kriging. *Mathematical geology* **20** (4), 405–421

De Loubens, C., Deschamps, J., Georgelin, M., Charrier, A., Edward-Lévy, F. & Leonetti, M. 2014 Mechanical characterization of cross–linked serum albumin microcapsules. *Soft Matter* **10**, 4561 – 4568.

De Vuyst, F., Dupont, C. & Salsac, A.-V. 2022 Space-time parameter PCA for data-driven modeling with application to bioengineering. in: Principal component analysis. *Ed: García Márquez F.P. Intech-Open* .

Demirgöz, D., Pangburn, T. O., Davis, K. P., Lee, S., Bates, F. S. & Kokkoli, E. 2009 Pr b-targeted delivery of tumor necrosis factor-α by polymersomes for the treatment of prostate cancer. *Soft Matter* **5** (10), 2011–2019.

Di Natale, C., Lagreca, E., Panzetta, V., Gallo, M., Passannanti, F., Vitale, M., Fusco, S., Vecchione, R., Nigro, R. & Netti, P. 2021 Morphological and rheological guided design for the microencapsulation process of lactobacillus paracasei cba l74 in calcium alginate microspheres. *Frontiers in bioengineering and biotechnology* **9**.

Diaz, A., Pelekasis, N. A. & Barthès-Biesel, D. 2000 Transient response of a capsule subjected to varying flow conditions : effect of internal fluid viscosity and membrane elasticity. *Phys. Fluids* **12**, 948–957.

Dikshit, H. P. & Powar, P. 1982 Discrete cubic spline interpolation. *Numerische Mathematik* **40** (1), 71–78.

Doddi, S. K. & Bagchi, P. 2008 Lateral migration of a capsule in a plane poiseuille flow in a channel. *Int. J. Multiphas. Flow* **34**, 966 – 986.

Dodson, W. R. & Dimitrakopoulos, P. 2008 Spindles, cusps, and bifurcation for capsules in Stokes flow. *Phys. Rev. Lett.* **101** (20), 208102.

Dodson, W. R. & Dimitrakopoulus, P. 2009 Dynamics of strain-hardening and strain-softening capsules in strong planar extensional flows via an interfacial spectral boundary element algorithm for elastic membranes. *Journal of Fluid Mechanics* **641**, 263–296.

Dubuisson, M.-P. & Jain, A. 1994 A modified hausdorff distance for object matching. In *Proceedings of 12th International Conference on Pattern Recognition*, , vol. 1, pp. 566–568 vol.1.

Dupont, C., De Vuyst, F. & Salsac, A.-V. 2021 A non-intrusive kinematics-consistent data-driven reduced-order model of deformable microcapsules in a stokes flow. *Journal of Fluid Mechanics (Submitted)* .

Erichson, N. B., Mathelin, L., Kutz, J. N. & Brunton, S. L. 2019 Randomized Dynamic Mode Decomposition. *SIAM Journal on Applied Dynamical Systems* **18** (4), 1867–1891

Fernandes, R. & Gracias, D. H. 2012 Self-folding polymeric containers for encapsulation and delivery of drugs. *Advanced drug delivery reviews* **64** (14), 1579–1589

Foessel, E., Walter, J., Salsac, A.-V. & Barthès-Biesel, D. 2011 Influence of internal viscosity on the large deformation and buckling of a spherical capsule in a simple shear flow. *J. Fluid Mech.* **672**, 477 – 486.

FRALEY, C. & RAFTERY, A. E. 1998 How many clusters? which clustering method? answers via model-based cluster analysis. *The computer journal* **41** (8), 578–588

GALLIVAN, K., VANDENDORPE, A. & VAN DOOREN, P. 2004 Model reduction of mimo systems via tangential interpolation. *SIAM Journal on Matrix Analysis and Applications* **26** (2), 328–349.

GHIRARDI, P., CATENAZZO, G., MANTERO, O., MEROTTI, G. C. & MARZO, A. 1977 Bioavailability of digoxin in a new soluble pharmaceutical formulation in capsules. *J. Pharm. Sci.* **66**, 267 – 269.

GHNATIOS, C., MASSON, F., HUERTA, A., LEYGUE, A., CUETO, E. & CHINESTA, F. 2012 Proper Generalized Decomposition based dynamic data-driven control of thermal processes. *Computer Methods in Applied Mechanics and Engineering* **213-216**, 29–41.

GOLUB, G. H. & REINSCH, C. 1971 *Singular Value Decomposition and Least Squares Solutions*, pp. 134–151. Berlin, Heidelberg: Springer Berlin Heidelberg.

GREEN, A. E. & ADKINS, J. E. 1970 *Large elastic deformations*, 2nd edn. Oxford University Press.

GREEN, A. E., ADKINS, J. E. & RIVLIN, R. S. 1971 Large elastic deformations. *Physics Today* **24** (12), 57–57.

HAM, J., LEE, D. D., MIKA, S. & SCHÖLKOPF, B. 2004 *A kernel view of the dimensionality reduction of manifolds*.

HASTIE, T., TIBSHIRANI, R., FRIEDMAN, J. H. & FRIEDMAN, J. H. 2009 *The elements of statistical learning: data mining, inference, and prediction*, , vol. 2. Springer.

HÉAS, P. & HERZET, C. 2017 Optimal kernel-based dynamic mode decomposition .

HEINRICH, V. & RAWICZ, W. 2005 Automated, high–resolution micropipet apsiration reveals new insight into the physical properties of fluid membranes. *Langmuir* **21**, 1962 – 1971.

HÖPPNER, F., KLAWONN, F., KRUSE, R. & RUNKLER, T. 1999 *Fuzzy cluster analysis: methods for classification, data analysis and image recognition*. John Wiley & Sons.

HRUSCHKA, E. R., CAMPELLO, R. J. & FREITAS, A. A. 2009 A survey of evolutionary algorithms for clustering. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* **39** (2), 133–155

HSU, C.-H., CHEN, C. & FOLCH, A. 2004 "microcanals" for micropipette access to single cells in microfluidic environments. *Lab on a Chip* **4** (5), 420–424.

Hu, X.-Q., Salsac, A.-V. & Barthès-Biesel, D. 2012 Flow of a spherical capsule in a pore with circular or square cross–section. *J. Fluid Mech.* **705**, 176 – 194.

Hu, X.-Q., Sévénié, B., Salsac, A.-V., Leclerc, E. & Barthès-Biesel, D. 2013 Characterization of membrane properties of capsules flowing in a square–section microfluidic channel: effects of the membrane constitutive law. *Phys. Rev. E.* **87**, 063008.

Iuliano, E. & Quagliarella, D. 2013 Aerodynamic shape optimization via non-intrusive pod-based surrogate modelling.

Jian, A. K. 2009 Data clustering: 50 years beyond k-means, pattern recognition letters. *Corrected Proof* .

Jiang, W., Li, M., Chen, Z. & Leong, K. W. 2016 Cell-laden microfluidic microgels for tissue regeneration. *Lab on a Chip* **16** (23), 4482–4506.

Kessler, S., Finken, R. & Seifert, U. 2008 Swinging and tumbling of elastic capsules in shear flow **605**, 207–226.

Kevrekidis, I., Rowley, C. W. & Williams, M. 2016 A kernel-based method for data-driven koopman spectral analysis. *Journal of Computational Dynamics* **2** (2), 247–265.

Kuriakose, S. & Dimitrakopoulos, P. 2011 Motion of an elastic capsule in a square microfluidic channel. *Phys. Rev. E* **84**, 011906.

Kwak, S. & Pozrikidis, C. 2001 Effect of membrane bending stiffness on the axisymmetric deformation of capsules in uniaxial extensional flow. *Physics of Fluids* **13** (5), 1234–1242.

Lac, E., Barthès-Biesel, D., Pelakasis, A. & Tsamopoulos, J. 2004 Spherical capsules in three–dimensional unbounded Stokes flows: effect of the membrane constitutive law and onset of buckling. *J. Fluid Mech.* **516**, 303 – 334.

Ladyzhenskaya, O. A. 1969 *The mathematical theory of viscous incompressible flow*, , vol. 2. Gordon and Breach New York.

Lancaster, P. & Salkauskas, K. 1981 Surfaces generated by moving least squares methods. *Mathematics of computation* **37** (155), 141–158, 0025–5718.

Lappano, E., Naets, F., Desmet, W., Mundo, D. & Nijman, E. 2016 A greedy sampling approach for the projection basis construction in parametric model order reduction for structural dynamics models. In *Proceedings of the 27th International Conference on Noise and Vibration Engineering.*, pp. 3563–3571.

Le Clainche, S. & Vega, J. M. 2017 Higher order dynamic mode decomposition. *SIAM Journal on Applied Dynamical Systems* **16** (2), 882–925, 1536–0040.

Lefebvre, Y. & Barthès-Biesel, D. 2007 Motion of a capsule in a cylindrical tube: effect of membrane pre-stress **589**, 157–181.

Lefebvre, Y., Leclerc, E., Barthès-Biesel, D., Walter, J. & Edwards-Levy, F. 2008 Flow of artificial microcapsules in microfluidic channels: A method for determining the elastic properties of the membrane. *Phys. Fluids* **20** (12), 123102.

Li, X. & Sarkar, K. 2008 Front tracking simulation of deformation and buckling instability of a liquid capsule enclosed by an elastic membrane. *Journal of Computational Physics* **227** (10), 4998–5018.

Li, X. Z., Barthès-Biesel, D. & Helmy, A. 1988 Large deformations and burst of a capsule freely suspended in an elongational flow **187**, 179 – 196.

Lumley, J. L. 1967 The structure of inhomogeneous turbulent flows. In *Atmospheric turbulence and radio propagation* (ed. A. M. Yaglom & V. I. Tatarski), pp. 166–178.

Luo, H. & Pozrikidis, C. 2007 Buckling of a pre-compressed or pre-stretched membrane in shear flow. *International Journal of Solids and Structures* **44** (24), 8074–8085.

Ma, G. & Su, Z.-G. 2013 *Microspheres and microcapsules in biotechnology: design, preparation and applications*. CRC Press.

Maxime, B., Yvon, M., Ngoc, C. N. & Anthony, T. P. 2004 An empirical interpolation method: application to efficient reduced-basis discretization of partial differential equations. *Comptes Rendus Mathematique* **339** (9), 667–672.

Muld, T. W., Efraimsson, G. & Henningson, D. S. 2012 Flow structures around a high-speed train extracted using proper orthogonal decomposition and dynamic mode decomposition. *Computers and Fluids* **57**, 87 – 97.

Nayroles, B., Touzot, G. & Villon, P. 1992 Generalizing the finite element method: Diffuse approximation and diffuse elements. *Computational Mechanics* **10** (5), 307–318.

Nazzaro, F., Orlando, P., Fratianni, F. & Coppola, R. 2012 Microencapsulation in food science and biotechnology. *Curr. Opin. Biotech.* **23**, 182 – 186.

Nielsen, F. 2016 *Introduction to HPC with MPI for Data Science*. Springer.

Noack, B. R., Stankiewicz, W., Morzyński, M. & Schmid, P. J. 2016 Recursive dynamic mode decomposition of transient and post-transient wake flows. *Journal of Fluid Mechanics* **809**, 843–872.

Onwulata, C. 2012 Encapsulation of new active ingredients. *Annual review of food science and technology* **3**, 183–202, 1941–1413.

Orive, G., Santos, E., Pedraz, J. L. & Hernández, R. M. 2014 Application of cell encapsulation for controlled delivery of biological therapeutics. *Adv. Drug Deliv. Rev.* **67-68**, 3 – 14.

Paolella, M. S. 2018 *Linear models and time-series analysis: regression, ANOVA, ARMA and GARCH*. John Wiley & Sons.

Papadrakakis, M., Lagaros, N. D. & Tsompanakis, Y. 1998 Structural optimization using evolution strategies and neural networks. *Computer Methods in Applied Mechanics and Engineering* **156** (1), 309–333.

Pawar, S., Rahman, S., Vaddireddy, H., San, O., Rasheed, A. & Vedula, P. 2019 A deep learning enabler for nonintrusive reduced order modeling of fluid flows. *Physics of Fluids* **31** (8), 085101, 1070–6631.

Pieper, G., Rehage, H. & Barthès-Biesel, D. 2005 Deformation of a capsule in a spinning drop apparatus. *J. Colloid Interface Sci.* **202**, 293–300.

Popel, A. S. & Johnson, P. C. 2005 Microcirculation and hemorheology. *Annual review of fluid mechanics* **37**, 43–69.

Pozrikidis, C. 1995 Finite deformation of liquid capsules enclosed by elastic membranes in simple shear flow. *J. Fluid Mech.* **297**, 123–152.

Pozrikidis, C. 2005 Numerical simulation of cell motion in tube flow. *Annu. Biomed. Eng.* **33**, 165 – 178.

Quesada, C., Dupont, C., Villon, P. & Salsac, A.-V. 2021 Diffuse approximation for identification of the mechanical properties of microcapsules. *Mathematics and Mechanics of Solids* **26** (7), 1018–1028.

RABANEL, J.-M., BANQUY, X., ZOUAOUI, H., MOKHTAR, M. & HILDGEN, P. 2009 Progress technology in microencapsulation methods for cell therapy. *Biotechnology Progress* **25** (4), 946–963.

RAGHAVAN, B., BREITKOPF, P., TOURBIER, Y. & VILLON, P. 2013 Towards a space reduction approach for efficient structural shape optimization. *Structural and Multidisciplinary Optimization* **48** (5), 987–1000.

RAISSI, M., PERDIKARIS, P. & KARNIADAKIS, G. 2019 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.

RAMANUJAN, S. & POZRIKIDIS, C. 1998 Deformation of liquid capsules enclosed by elastic membranes in simple shear flow: Large deformations and the effect of capsule viscosity. *J. Fluid Mech.* **361**, 117 – 143.

ROWLEY, C. W., MEZIĆ, I., BAGHERI, S., SCHLATTER, P. & HENNINGSON, D. S. 2009 Spectral analysis of nonlinear flows. *Journal of fluid mechanics* **641**, 115–127, 1469–7645.

RUGAR, D. & HANSMA, P. 1990 Atomic force microscopy. *Physics today* **43** (10), 23–30, 0031–9228.

SANDOVAL, S. & TOBIAS, G. 2021 Encapsulation of fullerenes: A versatile approach for the confinement and release of materials within open-ended multiwalled carbon nanotubes. *Frontiers in bioengineering and biotechnology* **9**.

SAVIGNAT, J.-M. 2000 Approximation diffuse Hermite et ses applications. Theses, École Nationale Supérieure des Mines de Paris.

SCHMID, P. J. 2010 Dynamic Mode Decomposition of numerical and experimental data. *Journal of fluid mechanics* **656**, 5–28, 1469–7645.

SECOMB, T. W. & SKALAK, R. 1982 A two-dimensional model for capillary flow of an asymmetric cell. *Microvascular Research* **24** (2), 194–203.

SEENA, A. & SUNG, H. J. 2011 Dynamic mode decomposition of turbulent cavity flows for self-sustained oscillations. *International Journal of Heat and Fluid Flow* **32** (6), 1098 – 1110.

SÉVÉNIÉ, B., SALSAC, A.-V. & BARTHÈS-BIESEL, D. 2015 Characterization of Capsule Membrane Properties using a Microfluidic Photolithographied Channel: Consequences of Tube Non-squareness. *Procedia IUTAM* **16**, 106–114.

SHE, F. Y., QI, D. M., ND J.Z. SHAO, Z. J. C. & YANG, L. 2012 Preparation of organic pigment microcapsules and its application in pigment printing of silk fabric. *Adv. Mat. Res.* **441**, 145 – 149.

SIEBER, M., PASCHEREIT, C. O. & OBERLEITHNER, K. 2016 Spectral proper orthogonal decomposition. *Journal of Fluid Mechanics* **792**, 798–828 , 0022–1120.

SILVA, D. F. & ALVARO, L. C. 2015 Practical implementation aspects of Galerkin reduced order models based on Proper orthogonal Decomposition for Computational Fluid Dynamics. *J. of Brasilian Society of Mechanical Sciences and Engineering* **37**, 1309–1327.

SINGH, M. N., HEMANT, K. S. Y., RAM, M. & SHIVAKUMAR, H. G. 2010 Microencapsulation: A promising technique for controlled drug delivery. *Research in pharmaceutical sciences* **5** (2), 65–77.

SIROVICH, L. 1987 Turbulence and the Dynamics of Coherent Structures. Part 1 : Coherent Structures. *Quarterly of Applied Mathematics* **45** (3), 561–571.

SOBEL, R. 2014 *Microencapsulation in the Food Industry: A Practical Implementation Guide*. Elsevier.

SUI, Y., LOW, H., CHEW, Y. & ROY, P. 2008 Tank-treading, swinging, and tumbling of liquid-filled elastic capsules in shear flow. *Physical Review E* **77** (1), 016310.

WALTER, J. 2009 Couplage intégrales de frontières – éléments finis : application aux capsules sphériques et ellipsoïdales en écoulement. Theses, Université de Technologie de Compiègne.

WALTER, J., SALSAC, A.-V. & BARTHES-BIESEL, D. 2011 Ellipsoidal capsules in simple shear flow: prolate versus oblate initial shapes. *Journal of Fluid Mechanics* **676**, 318–347, 1469–7645.

WALTER, J., SALSAC, A.-V., BARTHÈS-BIESEL, D. & LE TALLEC, P. 2010 Coupling of finite element and boundary integral methods for a capsule in a Stokes flow. *Int. J. Num. Meth. Engng* **83**, 829–850.

WANG, X.-Y., MERLO, A., DUPONT, C., SALSAC, A.-V. & BARTHES-BIESEL, D. 2021 A microfluidic methodology to identify the mechanical properties of capsules: comparison with a microrheometric approach. *Flow* **1**, 2633–4259.

WHANG, J. J., HOU, Y., GLEICH, D. F. & DHILLON, I. S. 2019 Non-exhaustive, overlapping clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **41** (11), 2644–2659.

Williams, M. O., Kevrekidis, I. G. & Rowley, C. W. 2015 A data–driven approximation of the koopman operator: Extending dynamic mode decomposition **25** (6), 1307–1346.

Wu, M. & Martin, M. P. 2007 Direct numerical simulation of supersonic turbulent boundary layer over a compression ramp. *AIAA Journal* **45** (4), 879–889.

Xiao, D., Fang, F., Buchan, A., Pain, C., Navon, O., Du, J. & Hu, G. 2014 Non-linear model reduction for the navier-stokes equations using residual DEIM method. *J. Comp. Phys.* **263**, 1–18.

Xiao, D., Fang, F., Buchan, A. G., Pain, C. C., Navon, I. M. & Muggeridge, A. 2015 Non-intrusive reduced order modelling of the navier–stokes equations. *Computer Methods in Applied Mechanics and Engineering* **293**, 522–541.

Xiao, D., Fang, F., Pain, C. & Navon, I. 2017 A parameterized non-intrusive reduced order model and error analysis for general time-dependent nonlinear partial differential equations and its applications. *Computer Methods in Applied Mechanics and Engineering* **317**, 868–889.

Yang, Z., Peng, Z., Li, J., Li, S., Kong, L., Li, P. & Wang, Q. 2014 Development and evaluation of novel flavour microcapsules containing vanilla oil using complex coacervation approach. *Food Chem.* **145**, 272 – 277.