A Data-Driven Space-Time-Parameter Reduced-Order Model with Manifold Learning for Coupled Problems: Application to Deformable Capsules Flowing in Microchannels

Toufik Boubehziz [1], Carlos Quesada-Granja [1], Claire Dupont [1], Pierre Villon [2], Florian De Vuyst [3,*] and Anne-Virginie Salsac [1]

1   Biomechanics and Bioengineering Laboratory (UMR CNRS 7338), Université de Technologie de Compiègne CNRS, Alliance Sorbonne Université, 60203 Compiègne, France; toufik.boubehziz@utc.fr (T.B.); quesadagranja@gmail.com (C.Q.-G.); claire.dupont@utc.fr (C.D.); a.salsac@utc.fr (A.-V.S.)
2   Laboratoire Roberval, Université de Technologie de Compiègne, Alliance Sorbonne Université, 60203 Compiègne, France; pierre.villon@utc.fr
3   Laboratoire de Mathématiques Appliquées de Compiègne (EA 2222), Université de Technologie de Compiègne, Alliance Sorbonne Université, 60203 Compiègne, France
*   Correspondence: florian.de-vuyst@utc.fr

Abstract: An innovative data-driven model-order reduction technique is proposed to model dilute micrometric or nanometric suspensions of microcapsules, i.e., microdrops protected in a thin hyperelastic membrane, which are used in Healthcare as innovative drug vehicles. We consider a microcapsule flowing in a similar-size microfluidic channel and vary systematically the governing parameter, namely the capillary number, ratio of the viscous to elastic forces, and the confinement ratio, ratio of the capsule to tube size. The resulting space-time-parameter problem is solved using two global POD reduced bases, determined in the offline stage for the space and parameter variables, respectively. A suitable low-order spatial reduced basis is then computed in the online stage for any new parameter instance. The time evolution of the capsule dynamics is achieved by identifying the nonlinear low-order manifold of the reduced variables; for that, a point cloud of reduced data is computed and a diffuse approximation method is used. Numerical comparisons between the full-order fluid-structure interaction model and the reduced-order one confirm both accuracy and stability of the reduction technique over the whole admissible parameter domain. We believe that such an approach can be applied to a broad range of coupled problems especially involving quasistatic models of structural mechanics.

Keywords: data-driven model; model order reduction; proper orthogonal decomposition; manifold learning; diffuse approximation; microcapsule suspension; Hausdorff distance

1. Introduction

Numerical modeling and simulation today appear to be an indispensable science to analyze physics-coupled problems (e.g., micrometric and nanometric suspensions), but also for innovative design and optimization of complex three-dimensional systems in engineering and industry (health, automotive, aircraft, etc.). Although one can nowadays find robust and accurate open-source or commercial codes for the simulation of multiphysics systems, it is still hard to use them in the context of robust or optimal design because of the prohibitive computational time that does not match with engineering production horizons. In order to accelerate computations, one can make use of parallel HPC (High Performance Computing) facilities, but this can become financially costly. For most applications, even with HPC facilities, the evaluation of the solutions takes days or weeks for three-dimensional multi-coupled problems.

Alternatively, a current tendency is to use machine learning or artificial intelligence tools to capitalize knowledge stored into data and use it for future case studies. It leads to

less redundant or useless computations, while the database of results continues to grow with more and more relevant information contents. With machine learning, one can expect to explore the design spaces in an easier, faster and more efficient way. However, one usually needs expertise to design and train artificial neural networks (ANN) correctly. For high-dimensional data, the training stage may require large computational resources and issues of quality of data may also be raised. Machine learning can be extended to time dependent problems and dynamical system using e.g., recurrent neural networks [1].

However, for particular use cases like physical systems, machine learning algorithms are designed to return three-dimensional spatial fields. This means that the outputs of the networks are high-dimensional vectors, which may induce training convergence and accuracy issues. Over the last two years, one could observe the rise of so-called Physics-informed neural networks (PINN) where the artificial neural networks are trained from a loss function that includes physical information (like partial differential equations), see e.g., [2].

Another class of 'machine learning' methods are the data-driven model order reduction (MOR) techniques, that use data generated from a (time-consuming) high-fidelity solver, also called the full-order model (FOM) [3,4]. Data-driven and non-intrusive reduced-order models (ROM) can be seen as supervised ANN [5,6]. For parametrized partial differential problems, ROMs usually perform a dimensionality reduction by means of suitable reduced bases. This can be achieved via different approaches such as the Proper Orthogonal Decomposition (POD) [7–9], piecewise tangential interpolation [10], Proper Generalized Decompositions (PGD) [11–13], Empirical Interpolation Methods (EIM) [14–16] or via different greedy procedures [17]. Then one has to find the manifold that maps the parameters to the coefficients of linear combination of the reduced basis functions. This can also be achieved in a supervised way, by means of universal approximation techniques like diffuse approximation [18] for example. Using ROMs may lead to substantial speedups as compared to FOM, from say 10 up to 10,000. One can even imagine real-time computations in some cases [19].

The 'ultimate' case is that of space-time-parameter problems involving spatial fields, timeline and design variables. This is of course of industrial importance, but still an issue and a current active field of research (see [20] for example). For such problems, the data are generally organized in data cubes (Figure 1). In this paper, a data-driven reduced-order modeling approach is proposed for space-time-parameter mechanical problems involving an equation of kinematics and a quasi-static law of equilibrium. As particular application, the physical problem that is addressed is the dynamics of dilute suspensions of micrometric capsules in microfluidic channels. Microcapsules can be used in Healthcare as innovative drug transportation vehicles into blood vessels and are expected to deliver drugs at identified targets [21,22]. They are composed of an elastic membrane protecting a liquid inner core and are used in suspension in another liquid. Testing them in microfluidic environments offers great potential to determine the capsule behavior and characterize the mechanical properties of the membrane [23–29], but also for sorting or enrichment of capsule suspensions [30–34]. We presently focus on the flow of a dilute suspension of initially spherical micrometric capsules in a microfluidic channel, which is a complex three-dimensional inertialess fluid-structure interaction problem that interestingly depends on only two independent design variables: the capillary number of the capsule Ca, which 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 the confinement ratio a/ℓ that provides a comparison between the initial capsule diameter and the channel width.
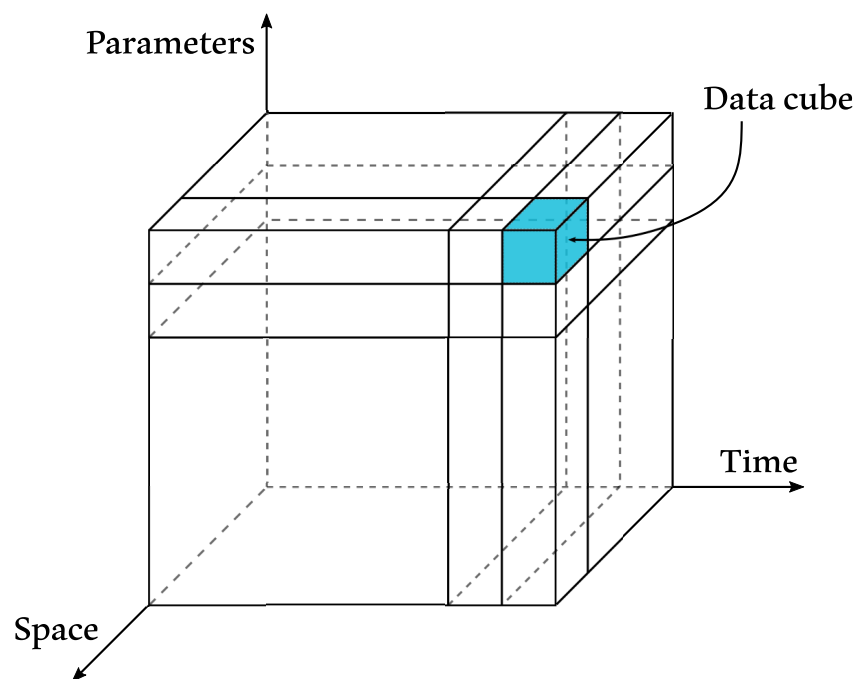
Figure 1. Space-time-parameter data cube.

Proper Orthogonal Decomposition has been shown to be particularly suitable to build reduced order models (ROM) of microcapsules [35], 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 [18,36,37] using the principal modes of both fields obtained by POD decomposition.

Numerical experiments will demonstrate the accuracy and efficiency of the approach, comparing reduced-order solutions to the full-order ones. We believe that the methodology proposed in this paper 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.

This paper is organized as follows. In Section 2, we describe the physical problem and its full order model solution. In Section 3.2, we construct parametric and spatial reduced-order modes using a set of pre-computed simulations. This allows to introduce a reduced-order model that expresses the displacement and velocity of a capsule at a selection of snapshots. In Section 3.4, we build a reduced model that corresponds to any parameter vector of Ca and a/ℓ values by estimating its corresponding principal components. 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. Finally, in Section 4, we will validate the whole computational ROM approach with a comparison to the FOM solutions.

2. Material and Methods

2.1. Problem Statement

Let us consider a spherical micrometric capsule of radius a freely placed in a three-dimensional microfluidic channel with square cross-section of length 2ℓ (see Figure 2). 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.
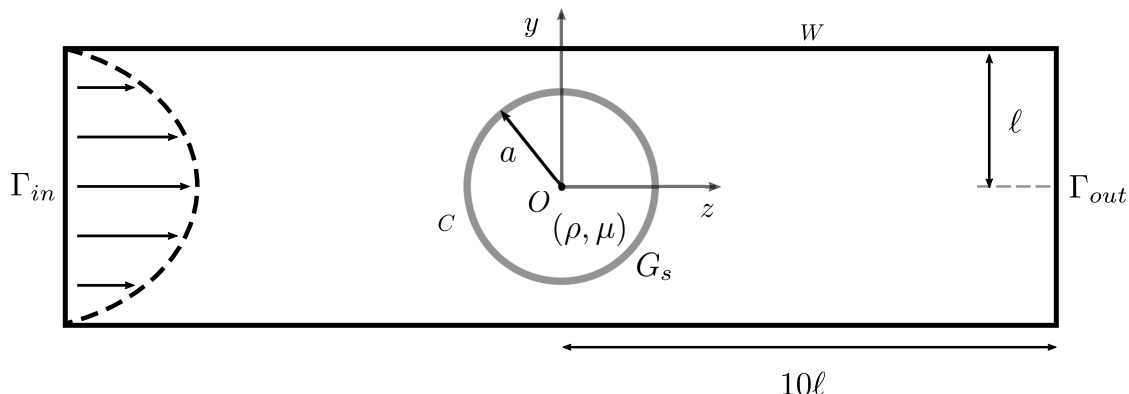

Figure 2. Initial configuration considered in the FOM model: an initially spherical capsule is placed at the center of a square-section channel. The time-evolution of its dynamics is computed using a reference frame centred onto the capsule centre of mass.

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 Reynolds number of the external flow is assumed to be very small (typically of order 10^{-2} or less), inertia being negligible because of the spatial scales involved in the problem. As far as the internal flow is concerned, its velocity is induced by the motion of the capsule membrane, which is itself entrained by the external flow: it is thus of smaller amplitude than that of the external flow. Hence, the flow in the internal ($\beta = in$) and external ($\beta = ex$) fluids are described by the Stokes equations:

$$\nabla \cdot v^\beta = 0, \ \nabla \cdot \sigma^\beta = 0, \ \beta = in, ex. \tag{1}$$

where σ^β is stress tensor in the fluids. The Stokes equations are defined in the domains bounded by the capsule membrane for $\beta = in$ and by the capsule membrane 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 in the FOM model). 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 v^β and the pressure field p^β, we consider the following boundary conditions (note that the boundary conditions include wall confinement effects, see [38] for more details):

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

$$v^{ex}(x, t) \rightarrow v^\infty(x), \text{ when } x \in \Gamma_{in} \cup \Gamma_{out}, \tag{2}$$

where v^∞ is the Poiseuille flow velocity of the suspending fluid in the absence of capsule. For a square channel we have the expression in expansion form

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

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

$$p^{ex}(\boldsymbol{x}, t) = 0, \; \boldsymbol{x} \in \Gamma_{in}, \tag{4}$$

$$p^{ex}(\boldsymbol{x}, t) = \Delta p(t) + \Delta p^{\infty}, \; \boldsymbol{x} \in \Gamma_{out}, \tag{5}$$

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 boundary conditions on the channel wall W:

$$\boldsymbol{v}^{ex}(\boldsymbol{x}, t) = 0, \; \text{for } \boldsymbol{x} \in W \tag{6}$$

- No slip boundary conditions on the capsule membrane C:

$$\boldsymbol{v}^{in}(\boldsymbol{x}, t) = \boldsymbol{v}^{ex}(\boldsymbol{x}, t) = \frac{\partial}{\partial t}\boldsymbol{x}(\boldsymbol{X}, t), \; \text{for } \boldsymbol{x} \in C \tag{7}$$

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

$$\left[\boldsymbol{\sigma}^{ex}(\boldsymbol{x}) - \boldsymbol{\sigma}^{in}(\boldsymbol{x})\right] \cdot \boldsymbol{n} = \boldsymbol{q}, \; \text{for } \boldsymbol{x} \in C \tag{8}$$

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, that measure the in-plane deformation. Owing to the combined effects of hydrodynamic forces, boundary confinement, and membrane deformability, the capsule can be highly deformed. Consequently, the choice of membrane constitutive law is important. 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 proteic membrane [23,24,39]. 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:

$$\tau_1 = \frac{G_s}{\lambda_1 \lambda_2}\left[\lambda_1^2 - \frac{1}{(\lambda_1 \lambda_2)^2}\right], \; \tau_2 = \frac{G_s}{\lambda_1 \lambda_2}\left[\lambda_2^2 - \frac{1}{(\lambda_1 \lambda_2)^2}\right]. \tag{9}$$

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 (2)–(8) with the Finite Element Method (FEM) that solves the membrane mechanical problem [38,40] 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 $\{\boldsymbol{v}\} = \{\mathcal{N}\}(\{\boldsymbol{u}\})$. 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

$$\frac{d}{dt}\{\boldsymbol{u}\}(t) = \{\boldsymbol{v}\}(t),$$
$$\{\boldsymbol{v}\}(t) = \{\mathcal{N}\}(\{\boldsymbol{u}\}(t), \boldsymbol{\theta}), \quad t \in (0, T_f],$$
$$\{\boldsymbol{u}\}(0) = \{\boldsymbol{0}\}, \ \{\boldsymbol{v}\}(t) = \{\mathcal{N}\}(\{\boldsymbol{0}\}, \boldsymbol{\theta})$$

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

$$\{\boldsymbol{u}^{i+1}\} = \{\boldsymbol{u}^i\} + \delta t \{\boldsymbol{v}^i\},$$
$$\{\boldsymbol{v}^{i+1}\} = \{\mathcal{N}\}(\{\boldsymbol{u}^{i+1}\}, \boldsymbol{\theta}),$$
$$\{\boldsymbol{u}^0\} = \{\boldsymbol{0}\}, \ \{\boldsymbol{v}^0\} = \{\mathcal{N}\}(\{\boldsymbol{0}\}, \boldsymbol{\theta})$$

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:

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

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

$$\dot{\gamma}\,\delta t < C\,\frac{\Delta h_C}{\ell}\,Ca, \tag{10}$$

where $\dot{\gamma} = V/\ell$, $C > 0$ is a constant and Δh_C is the typical mesh size (see [40]). In practice, we first use small time steps, and tune them to be big enough but not too close to the stability boundary. This process does not take too much time.

2.3. Design of Experiment, Database of FOM Results

Simulations of the FOM problem have been run varying the governing parameters in the range $[0; 0.2]$ for the capillary number Ca and $[0.75; 1.2]$ for the confinement ratio a/ℓ. For $a/\ell \geq 0.95$, the capsule is initially pre-deformed into an ellipsoid of semi-minor axis equal to 0.9. This pre-deformation does not have any impact on the steady-state capsule shape and is enough to avoid contacts between the capsule membrane and the channel wall. The resulting numerical database, composed of $N_c = 118$ (Ca, a/l) samples (Figure 3), contains the time-evolution of the three-dimensional position (or displacements) vectors of the capsule membrane nodes. Only the configurations for which a steady-state shape has been reached are retained. No steady state is found above the dotted red line of Figure 3, the microcapsules exhibiting continuous elongation owing to the strain-softening behavior of the membrane law [41].
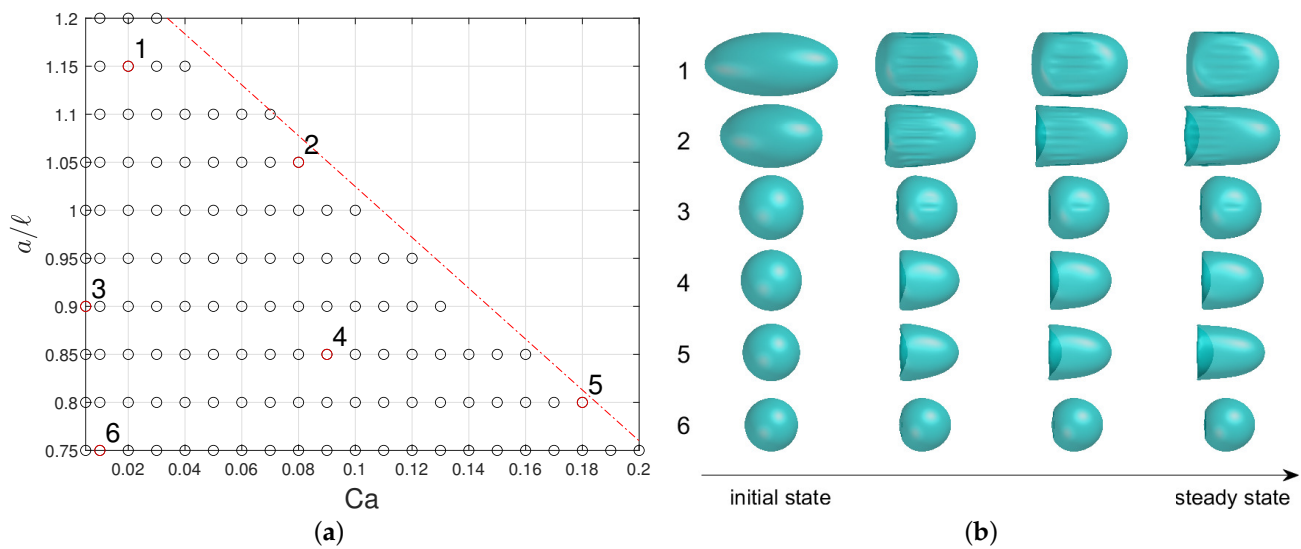
Figure 3. (**a**) Values of *Ca* and *a*/*l* included in the FOM database in the case of an initially spherical capsule with a Neo-Hookean membrane flowing in a square-section microfluidic channel. The parameter domain where a steady capsule deformation can be reached is delimited by the red dotted line. (**b**) Time evolution of capsule deformation along the microfluidic channel shown as illustration for the 6 cases indicated with numbers in figure (**a**). The capsule is pre-deformed into an ellipsoid when $a/\ell \geq 0.95$.

In the ROM model, we consider the capsule positions in the laboratory reference frame (and not the reference frame centred on the capsule centre of mass as in the FOM model). The capsule thus moves along the microchannel. For data generation, we pick up time snapshot solutions at coarser discrete times $t^i = i\,\Delta t$, where $\Delta t = m\,\delta t$ for some integer $m \geq 1$. The total number of coarse discrete times is denoted N_t. Let $\{X\}_{[n]}$ and $\{x\}_{[n]}(t^i) \in \mathbb{R}^3$, $n = 1, \ldots, N_x$, be the coordinates of node number n of the capsule mesh in the reference configuration (at time $t = 0$) and at discrete time t^i, $i = 1, \ldots, N_t$. The coordinates in the current configuration $\{x\}(t^i)$ are function of the $(Ca, a/\ell)$ parameter value denoted $\boldsymbol{\theta}_j$, $j = 1, \ldots, N_c$. The database is thus stored as a datacube of 3D-space, 1D-time and 2D-parameter data. The displacement vector is then $\{u\}\left(\{X\}_{[n]}, t^i, \boldsymbol{\theta}_j\right) = \{x\}_{[n]}(t^i) - \{X\}_{[n]}$. The velocity vector $\{v\}$ is calculated by finite differences from the position vector. Typically, for a standard capsule FOM simulation, N_x is of order 10^3 and N_c of order 10^2. The time step Δt is chosen such that N_t is of order 10^2.

3. Reduced Order Model (ROM)

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. Overview

We first give a general overview of the proposed data-driven model order reduction methodology. The approach is classically made of an offline stage for the search of the principal components and POD coefficient matrices of the FOM solutions, followed by an online stage where a parameter is chosen and a low-order dynamical system is run to get the solutions.

1. **Offline stage**. We reduce the data dimensionality by means of a double POD basis for space and parameter variables. The displacement field is represented as

    $$\{u\}(\{X\}, t, \boldsymbol{\theta}) = \sum_{k=1}^{K_u^x} \sum_{\ell=1}^{K_u^c} A_{k\ell}(t) \{\boldsymbol{\Phi}_u^r\}_k (\boldsymbol{\psi}_u(\boldsymbol{\theta}))_\ell, \tag{11}$$

    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:

    $$\{v\}(\{X\}, t, \boldsymbol{\theta}) = \sum_{k=1}^{K_v^x} \sum_{\ell=1}^{K_v^c} B_{k\ell}(t) \{\boldsymbol{\Phi}_v^r\}_k (\boldsymbol{\psi}_v(\boldsymbol{\theta}))_\ell. \tag{12}$$

    The determination of the double POD basis is achieved by singular value decomposition (SVD) from the datacube with different rearrangements of the data 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 accuracy of the representations.

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

    (a) Estimate the displacement field $\{u\}(\{X\}, t, \boldsymbol{\theta}_q)$ from expression (11). This requires an interpolation process at $\boldsymbol{\theta} = \boldsymbol{\theta}_q$. For that, we decide to use a diffuse approximation technique [18] that can be used for any parameter space dimension;

    (b) From the estimated displacement field $\{u\}(\{X\}, t^i, \boldsymbol{\theta}_q)$ computed at different instants $t^i \in [0, T_f]$, compute a low-order reduced basis $\{\boldsymbol{\varphi}^k\}(\boldsymbol{\theta}_q), k = 1, \ldots, m_u$ by singular value decomposition. We then get the low-order representations of both displacements and velocities:

    $$\{u\}(\{X\}, t, \boldsymbol{\theta}_q) = \sum_{k=1}^{m_u} \alpha_k(t) \{\boldsymbol{\varphi}^k\}(\boldsymbol{\theta}_q), \tag{13}$$

    $$\{v\}(\{X\}, t, \boldsymbol{\theta}_q) = \sum_{k=1}^{m_v} \xi_k(t) \{\boldsymbol{\gamma}^k\}(\boldsymbol{\theta}_q), \tag{14}$$

    (c) Manifold learning online stage: using diffuse approximation, we determine the low-order manifold $\mathcal{M}$ that links 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, easy to solve numerically: for $\boldsymbol{\theta} = \boldsymbol{\theta}_q$, solve

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

    The high-dimensional displacement and velocity fields can then be reconstructed according to (13) and (14).

    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.

Let $S_u^i \in \mathcal{M}_{3N_x, N_c}(\mathbb{R})$ be the matrix of capsule displacement fields $\{u\}$ and $S_v^i \in \mathcal{M}_{3N_x, N_c}(\mathbb{R})$ the matrix of the velocity fields $\{v\}$ at time t^i, $i = 1, \ldots, N_t$ (Figure 4a), considering all the configurations $\boldsymbol{\theta}_j$ for $j = 1, \ldots, N_c$ of the database, i.e.,

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

and

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

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

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

We then apply SVD [42] and get:

$$\mathcal{S}_\varrho = U_\varrho \, \Sigma_{S_\varrho} \, \Psi_\varrho^T, \text{ for } \varrho = u, v, \tag{15}$$

where $U_\varrho \in \mathcal{M}_{3N_x N_t, N_c}(\mathbb{R})$, $\Psi_\varrho \in \mathcal{M}_{N_c}(\mathbb{R})$ 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:

$$\begin{cases} \Psi_\varrho^r = \left[(\Psi_\varrho)_1, \ldots, (\Psi_\varrho)_{K_\varrho^c}\right] \in \mathcal{M}_{N_c, K_\varrho^c}(\mathbb{R}), \\ \text{with } (\Psi_\varrho)_k \in \mathcal{M}_{N_c, 1}(\mathbb{R}) \text{ for } k = 1, \ldots, K_\varrho^c, \text{ and } \varrho = u, v. \end{cases} \tag{16}$$

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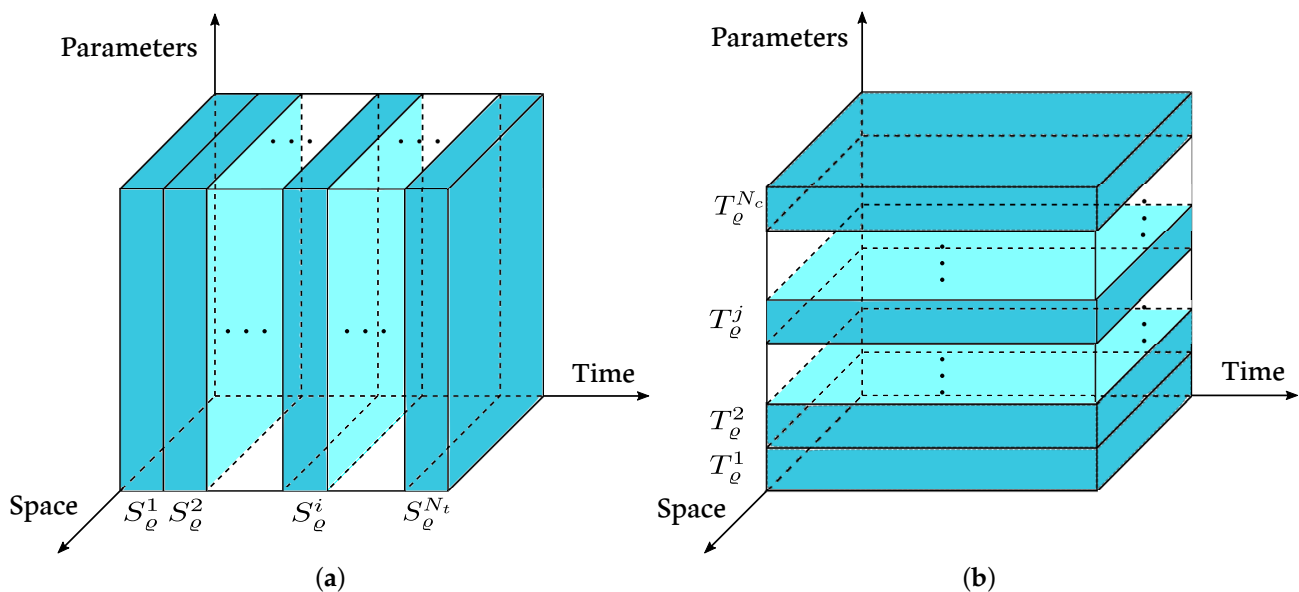

Figure 4. FOM data rearrangements for (**a**) parametric data set selection and (**b**) spatial data set selection.

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 $T_u^j \in \mathcal{M}_{3N_x,N_t}(\mathbb{R})$ be the displacement matrix and T_v^j the velocity matrix for the j-th configuration $\boldsymbol{\theta}_j$, for $j = 1,\ldots,N_c$ at all time instants t^i, $i = 1,\ldots,N_t$ (Figure 4b):

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

and

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

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 T_ϱ^j for $j = 1,\ldots,N_c$ and $\varrho = u,v$, respectively:

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

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

$$\mathcal{T}_\varrho = \Phi_\varrho \, \Sigma_{T_\varrho} \, V_\varrho^T, \text{ for } \varrho = u,v, \tag{17}$$

where $\Phi_\varrho \in \mathcal{M}_{3N_x}(\mathbb{R})$, $V_\varrho \in \mathcal{M}_{N_c N_t,3N_x}(\mathbb{R})$ 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:

$$\Phi_\varrho^r = \left[\{\boldsymbol{\phi}_\varrho\}_1,\ldots,\{\boldsymbol{\phi}_\varrho\}_{K_\varrho^x}\right] \in \mathcal{M}_{3N_x,K_\varrho^x}(\mathbb{R}) \tag{18}$$

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

$$\{\boldsymbol{u}\}\left(\{\boldsymbol{X}\}, t^i, [\boldsymbol{\theta}_1,\ldots,\boldsymbol{\theta}_{N_c}]\right) \approx \Phi_u^r \, A(t^i) \, (\Psi_u^r)^T \in \mathcal{M}_{3N_x,N_c}(\mathbb{R}), \tag{19}$$

$$\{\boldsymbol{v}\}\left(\{\boldsymbol{X}\}, t^i, [\boldsymbol{\theta}_1,\ldots,\boldsymbol{\theta}_{N_c}]\right) \approx \Phi_v^r \, B(t^i) \, (\Psi_v^r)^T \in \mathcal{M}_{3N_x,N_c}(\mathbb{R}), \tag{20}$$

where $A(t^i) \in \mathcal{M}_{K_u^x,K_u^c}(\mathbb{R})$ and $B(t^i) \in \mathcal{M}_{K_v^x,K_v^c}(\mathbb{R})$ 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 $i = 1\ldots,N_t$:

$$A(t^i) = \underbrace{(\Phi_u^r)^T}_{K_u^x \times (3N_x)} \, \underbrace{\{\boldsymbol{u}\}\left(\{\boldsymbol{X}\},t^i,[\boldsymbol{\theta}_1,\ldots,\boldsymbol{\theta}_{N_c}]\right)}_{(3N_x) \times N_c} \, \underbrace{\Psi_u^r}_{N_c \times K_u^c}, \tag{21}$$

$$B(t^i) = \underbrace{(\Phi_v^r)^T}_{K_v^x \times (3N_x)} \, \underbrace{\{\boldsymbol{v}\}\left(\{\boldsymbol{X}\},t^i,[\boldsymbol{\theta}_1,\ldots,\boldsymbol{\theta}_{N_c}]\right)}_{(3N_x) \times N_c} \, \underbrace{\Psi_v^r}_{N_c \times K_v^c}. \tag{22}$$

The outputs of the offline stage are respectively the POD matrices $\Phi_u^r, \Phi_v^r, \Psi_u^r, \Psi_v^r$ and the small matrices $A(t^i)$, $B(t^i)$, $i = 1,\ldots,N_t$. The next online stage will operate on the summarized data (19), (20) with coefficients matrices (21), (22). 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_ϱ^c, number of snapshots N_t.
  for $i \leftarrow 1, \ldots, N_t$ **do**
    if $(\varrho = u)$ **then**
      $S_u^i \leftarrow \left[\{\boldsymbol{u}\}(\{\boldsymbol{X}\}, t^i, \boldsymbol{\theta}_1), \; \ldots, \; \{\boldsymbol{u}\}(\{\boldsymbol{X}\}, t^i, \boldsymbol{\theta}_{N_c})\right]; \mathcal{S}_u \leftarrow [\mathcal{S}_u; S_u^i];$
    else
      $S_v^i \leftarrow \left[\{\boldsymbol{v}\}(\{\boldsymbol{X}\}, t^i, \boldsymbol{\theta}_1), \; \ldots, \; \{\boldsymbol{v}\}(\{\boldsymbol{X}\}, t^i, \boldsymbol{\theta}_{N_c})\right]; \mathcal{S}_v \leftarrow [\mathcal{S}_v, S_v^i];$
    end if
  end for
  for $j \leftarrow 1, \ldots, N_c$ **do**
    if $(\varrho = u)$ **then**
      $T_u^j \leftarrow \left[\{\boldsymbol{u}\}(\{\boldsymbol{X}\}, t^1, \boldsymbol{\theta}_j), \; \ldots, \; \{\boldsymbol{u}\}(\{\boldsymbol{X}\}, t^{N_t}, \boldsymbol{\theta}_j)\right]; \mathcal{T}_u \leftarrow [\mathcal{T}_u, T_u^j];$
    else
      $T_v^j \leftarrow \left[\{\boldsymbol{v}\}(\{\boldsymbol{X}\}, t^1, \boldsymbol{\theta}_j), \; \ldots, \; \{\boldsymbol{v}\}(\{\boldsymbol{X}\}, t^{N_t}, \boldsymbol{\theta}_j)\right]; \mathcal{T}_v \leftarrow [\mathcal{T}_v, T_v^j];$
    end if
  end for
  $\Phi_\varrho \leftarrow \text{SVD}(\mathcal{S}_\varrho), \Psi_\varrho \leftarrow \text{SVD}(\mathcal{T}_\varrho), \text{ for } \varrho \leftarrow u, v;$
  for $i = 1, \ldots, N_t$ **do**
    $A(t^i) \leftarrow (\Phi_u^r)^T \{\boldsymbol{u}\}(\{\boldsymbol{X}\}, t^i, [\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_{N_c}]) \Psi_u^r;$
    $B(t^i) \leftarrow (\Phi_v^r)^T \{\boldsymbol{v}\}(\{\boldsymbol{X}\}, t^i, [\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_{N_c}]) \Psi_v^r;$
  end for

3.4. Online Stage: Search for an Approximate Solution at a Query Configuration $\boldsymbol{\theta}_q$

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 $\{\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_{N_c}\}$, we consider a polynomial Lagrange interpolation with Lagrange polynomials denoted by $L_i(\boldsymbol{\theta})$ such that the Lagrange property

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

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

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

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 [18,29] 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. It is particularly suited for the current problem, for which the sampling is performed on a Cartesian grid. It may fail in the occurrence of local point alignment within the cloud points, which does not occur in the present study. The accuracy of the DA method may slightly decrease close to the boundary of the domain, the number of neighboring points being reduced.

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

$$\{\boldsymbol{u}\}(\{\boldsymbol{x}\}, t^i, \boldsymbol{\theta}_q) = \Phi_u^r A(t^i) \, \boldsymbol{\psi}_u(\boldsymbol{\theta}_q) \tag{23}$$

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

$$\{\boldsymbol{v}\}(\{\boldsymbol{x}\}, t^i, \boldsymbol{\theta}_q) = \Phi_v^r B(t^i) \, \boldsymbol{\psi}_v(\boldsymbol{\theta}_q). \tag{24}$$

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

$$\Psi_\varrho(\boldsymbol{\theta}_q) = \mathcal{A}_\varrho \, \boldsymbol{p}(\boldsymbol{\theta}_q), \text{ for } \varrho = u, v, \tag{25}$$

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

$$\begin{cases} \boldsymbol{p}(\boldsymbol{\theta}) = \begin{bmatrix} 1 & Ca & a/\ell \end{bmatrix}^T, \, m = 3 & \text{for first order DA,} \\ \boldsymbol{p}(\boldsymbol{\theta}) = \begin{bmatrix} 1 & Ca & a/\ell & Ca(a/\ell) & (Ca)^2 & (a/\ell)^2 \end{bmatrix}^T, \, m = 6 & \text{for second order DA.} \end{cases} \tag{26}$$

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 5a). 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 5a) 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

$$d = \left(\left(\theta_1 - (\boldsymbol{\theta}_q)_1\right)^2 + \tilde{r}^2\left(\theta_2 - (\boldsymbol{\theta}_q)_2\right)^2\right)^{\frac{1}{2}} / R. \tag{27}$$

The compactly supported Wendland weight function shown in Figure 5b is classically used. It has appropriate high-order approximation properties ([43]):

$$\begin{cases} w(d) = 2\,d^3 - 3\,d^2 + 1, & d \leq 1, \\ 0, & \text{otherwise.} \end{cases} \tag{28}$$

Diffuse approximation consists in minimizing the weighted least square problem

$$\min_{\mathcal{A}_\varrho \in \mathcal{M}_{K_\varrho^c, m}(\mathbb{R})} \quad J_{\boldsymbol{\theta}_q}(\mathcal{A}_\varrho) := \frac{1}{2} \sum_{\boldsymbol{\theta} \in \mathscr{S}(\boldsymbol{\theta}_q)} w(d(\boldsymbol{\theta})) \left\| \mathcal{A}_\varrho \, \boldsymbol{p}(\boldsymbol{\theta}) - [\Psi_\varrho^r(\boldsymbol{\theta})]^T \right\|^2_{\mathbb{R}^{K_\varrho^c}} \tag{29}$$

where $[\Psi_\varrho^r(\boldsymbol{\theta})]^T$ 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 (29) is then

$$\mathcal{A}_\varrho = (\Psi_\varrho^r)^T \mathcal{W} \mathcal{P} \left(\mathcal{P}^T \mathcal{W} \mathcal{P}\right)^{-1} \in \mathcal{M}_{K_\varrho^c, m}(\mathbb{R}) \tag{30}$$

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

$$\mathcal{P} = \begin{bmatrix} p(\boldsymbol{\theta}_1)^T \\ \vdots \\ p(\boldsymbol{\theta}_{N_c})^T \end{bmatrix} \text{ and } \mathcal{W} = \begin{bmatrix} w_1 & 0 & \cdots & 0 \\ 0 & w_2 & & \vdots \\ \vdots & & \ddots & \\ 0 & \cdots & & w_{N_c} \end{bmatrix}. \tag{31}$$
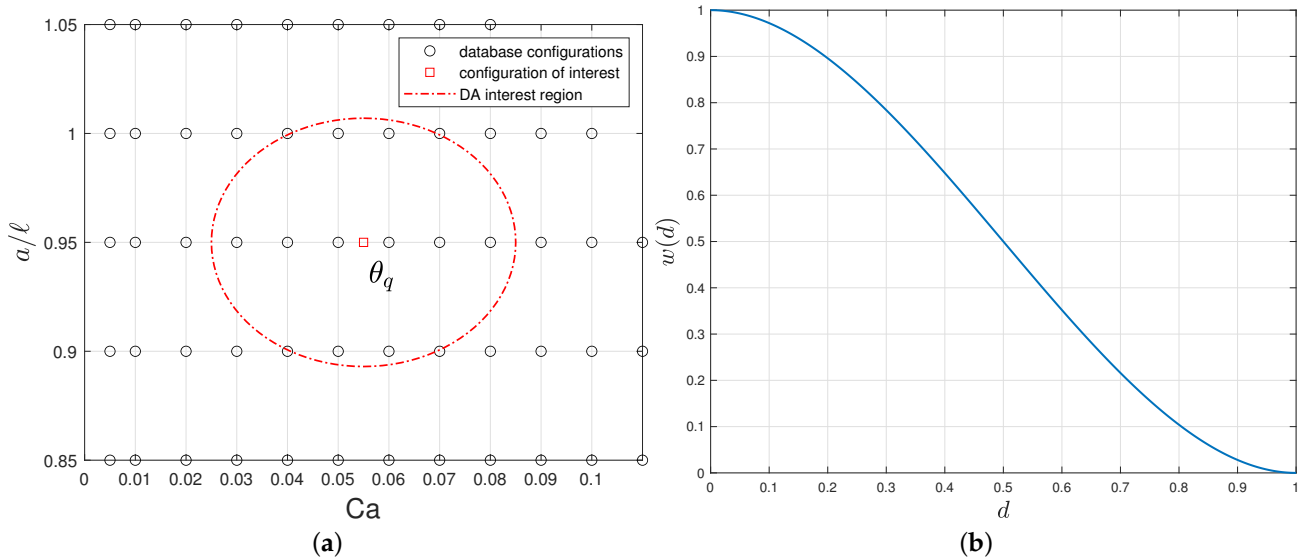

Figure 5. (**a**) DA elliptical region of interest (dashed line) defined around the point $\boldsymbol{\theta}_q = (Ca = 0.055, a/\ell = 0.95)$ in the parametric space with $M = 10$ neighbors; (**b**) Weight function $w(d)$.

3.4.2. Construction of a Low-Order Reduced Basis Suitable for $\boldsymbol{\theta} = \boldsymbol{\theta}_q$, Data Generation

From (23) and (24), 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:

$$\begin{cases} \mathcal{U}(\boldsymbol{\theta}_q) = \left[\{u\}(\{X\}, t^1, \boldsymbol{\theta}_q), \dots, \{u\}(\{X\}, t^{N_t}, \boldsymbol{\theta}_q) \right], \\ \mathcal{V}(\boldsymbol{\theta}_q) = \left[\{v\}(\{X\}, t^1, \boldsymbol{\theta}_q), \dots, \{v\}(\{X\}, t^{N_t}, \boldsymbol{\theta}_q) \right]. \end{cases} \tag{32}$$

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 $\{\boldsymbol{\varphi}^k\}(\boldsymbol{\theta}_q) \in \mathbb{R}^{3N_x}, k = 1, \dots, m_u$ for $\{u\}$ and velocity POD modes $\{\boldsymbol{\gamma}^k\}(\boldsymbol{\theta}_q) \in \mathbb{R}^{3N_x}, k = 1, \dots, m_v$ for $\{v\}$.

$$\text{POD}(\mathcal{U}(\boldsymbol{\theta}_q)) \quad \to \quad \{\boldsymbol{\varphi}^1\}(\boldsymbol{\theta}_q), \dots, \{\boldsymbol{\varphi}^{m_u}\}(\boldsymbol{\theta}_q) \tag{33}$$

$$\text{POD}(\mathcal{V}(\boldsymbol{\theta}_q)) \quad \to \quad \{\boldsymbol{\gamma}^1\}(\boldsymbol{\theta}_q), \dots, \{\boldsymbol{\gamma}^{m_v}\}(\boldsymbol{\theta}_q) \tag{34}$$

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

$$\{u\}(\{X\}, t, \boldsymbol{\theta}_q) = \sum_{k=1}^{m_u} \alpha_k(t) \{\boldsymbol{\varphi}^k\}(\boldsymbol{\theta}_q), \tag{35}$$

$$\{v\}(\{X\}, t, \boldsymbol{\theta}_q) = \sum_{k=1}^{m_v} \xi_k(t) \{\boldsymbol{\gamma}^k\}(\boldsymbol{\theta}_q). \tag{36}$$

By denoting

$$\Phi(\boldsymbol{\theta}_q) = \left[\{\boldsymbol{\varphi}^1\}(\boldsymbol{\theta}_q), \ldots, \{\boldsymbol{\varphi}^{m_u}\}(\boldsymbol{\theta}_q) \right] \in \mathcal{M}_{3Nx, m_u}(\mathbb{R}), \tag{37}$$

$$\Gamma(\boldsymbol{\theta}_q) = \left[\{\boldsymbol{\gamma}^1\}(\boldsymbol{\theta}_q), \ldots, \{\boldsymbol{\gamma}^{m_v}\}(\boldsymbol{\theta}_q) \right] \in \mathcal{M}_{3Nx, m_v}(\mathbb{R}) \tag{38}$$

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

$$\{\boldsymbol{u}\}(\{\boldsymbol{X}\}, t, \boldsymbol{\theta}_q) = \Phi(\boldsymbol{\theta}_q)\,\boldsymbol{\alpha}(t), \quad \{\boldsymbol{v}\}(\{\boldsymbol{X}\}, t, \boldsymbol{\theta}_q) = \Gamma(\boldsymbol{\theta}_q)\,\boldsymbol{\xi}(t). \tag{39}$$

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

$$\{\boldsymbol{u}^{i+1}\} = \{\boldsymbol{u}^i\} + \delta t^{ROM}\{\boldsymbol{v}^i\}, \tag{40}$$

$$\{\boldsymbol{v}^{i+1}\} = \{\mathcal{N}\}(\{\boldsymbol{u}^{i+1}\}, \boldsymbol{\theta}_q). \tag{41}$$

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 (40), (41) but formulated as a low-dimensional system. If $\{\boldsymbol{u}^i\}$ and $\{\boldsymbol{v}^i\}$ are searched in the form $\{\boldsymbol{u}^i\} = \Phi(\boldsymbol{\theta}_q)\,\boldsymbol{\alpha}^i$ and $\{\boldsymbol{v}^i\} = \Gamma(\boldsymbol{\theta}_q)\,\boldsymbol{\xi}^i$, respectively, Equation (40) becomes

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

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

$$\boldsymbol{\alpha}^{i+1} = \boldsymbol{\alpha}^i + \delta t^{ROM}\,Q(\boldsymbol{\theta}_q)\,\boldsymbol{\xi}^i, \tag{42}$$

where $Q(\boldsymbol{\theta}_q) = [\Phi(\boldsymbol{\theta}_q)]^T \Gamma(\boldsymbol{\theta}_q)$. Equation (41) is replaced by

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

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

$$\mathcal{M}(\boldsymbol{\alpha}^{i+1}, \boldsymbol{\theta}_q) = [\Gamma(\boldsymbol{\theta}_q)]^T \{\mathcal{N}\}(\Phi(\boldsymbol{\theta}_q)\,\boldsymbol{\alpha}^{i+1}, \boldsymbol{\theta}_q) \in \mathbb{R}^{m_v}. \tag{43}$$

3.4.4. Manifold Learning

Because of nonlinear terms, the direct computation of $\mathcal{M}(\boldsymbol{\alpha}^{i+1}, \boldsymbol{\theta}_q)$ in (43) 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 $\alpha_k^D(t^i)$, $k = 1, \ldots, m_u$, $i = 1, \ldots, N_t$ to the output data $\xi_k^D(t^i)$, $k = 1, \ldots, m_v$, $i = 1, \ldots, N_t$ ('*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

$$\xi_k = \boldsymbol{p}(\boldsymbol{\alpha})^T \boldsymbol{a}^k, \quad k = 1, \dots, m_v \tag{44}$$

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 $k = 1, \dots, m_v$, one looks for a coefficient vector $\boldsymbol{a}^k(t) \in \mathbb{R}^{m_u+1}$ solution of the weighted least square problem

$$\boldsymbol{a}^k(t) = \arg \min_{\boldsymbol{a} \in \mathbb{R}^{m_u+1}} \quad \frac{1}{2} \sum_{i=1}^{N_t} w\left(\frac{|t - t^i|}{R}\right) \left(\boldsymbol{p}(\boldsymbol{\alpha}^D(t^i))^T \boldsymbol{a} - \xi_k^D(t^i)\right)^2 \tag{45}$$

where $t \in [0, T_f]$, $w = w(d)$ is the weight function defined in Figure 5b and $d = \frac{|t-t^i|}{R}$. This returns a regression function

$$\xi_k = \xi_k(t, \boldsymbol{\alpha}(t)) = \boldsymbol{p}(\boldsymbol{\alpha}(t))^T \boldsymbol{a}^k(t). \tag{46}$$

3.4.5. Low-Order Dynamical Reduced Order Model

The resulting time-discrete reduced-order model is then

$$t^{i+1,ROM} = t^{i,ROM} + \delta t^{ROM}, \tag{47}$$

$$\boldsymbol{\alpha}^{i+1} = \boldsymbol{\alpha}^i + \delta t^{ROM} Q(\boldsymbol{\theta}_q) \boldsymbol{\xi}^i, \tag{48}$$

$$\xi_k^{i+1} = \boldsymbol{p}(\boldsymbol{\alpha}^{i+1})^T \boldsymbol{a}^k(t^{i+1,ROM}) \quad \forall k \in \{1, \dots, m_v\}. \tag{49}$$

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

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

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 $\Psi_u(\boldsymbol{\theta}_q)$ and $\Psi_v(\boldsymbol{\theta}_q)$ from the diffuse approximation approach;
   for $i = 1 \dots, N_t$ **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
   $\mathcal{U}(\boldsymbol{\theta}_q) \leftarrow [\{\boldsymbol{u}\}(\{\boldsymbol{x}\}, t^1, \boldsymbol{\theta}_q), \dots, \{\boldsymbol{u}\}(\{\boldsymbol{x}\}, t^{N_t}, \boldsymbol{\theta}_q)]$;
   $\mathcal{V}(\boldsymbol{\theta}_q) \leftarrow [\{\boldsymbol{v}\}(\{\boldsymbol{x}\}, t^1, \boldsymbol{\theta}_q), \dots, \{\boldsymbol{v}\}(\{\boldsymbol{x}\}, t^{N_t}, \boldsymbol{\theta}_q)]$;
   Compute $\Phi(\boldsymbol{\theta}_q), \Gamma(\boldsymbol{\theta}_q), Q(\boldsymbol{\theta}_q), \boldsymbol{\alpha}^D(t^i)$ and $\boldsymbol{\xi}^D(\boldsymbol{\theta}_i), i = 1, \dots, N_t$;
   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 $\boldsymbol{a}^k(t^{i+1,ROM})$, $k = 1, \dots, m_v$ 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:
     $\{\boldsymbol{u}\}(\{\boldsymbol{X}\}, t^{i+1,ROM}, \boldsymbol{\theta}_q) = \Phi(\boldsymbol{\theta}_q) \boldsymbol{\alpha}^{i+1}$;
     $\{\boldsymbol{v}\}(\{\boldsymbol{X}\}, t^{i+1,ROM}, \boldsymbol{\theta}_q) = \Gamma(\boldsymbol{\theta}_q) \boldsymbol{\xi}^{i+1}$;
   end while

---

4. Numerical Experiments

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 [38,40] is then used as FOM solver. The comparison of the FOM results with experimental ones using a square-base cylinder have been thoroughly described in other previous studies (see for example [23,24,27,39] from A.V. Salsac's research team). 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 6). A second-order RK2 Ralston scheme is used for time integration. The dimensionless time step is $\dot{\gamma}\delta t = 5 \cdot 10^{-4}$ for $Ca > 0.02$ and $\dot{\gamma}\delta t = 10^{-4}$ for $Ca \leq 0.02$.
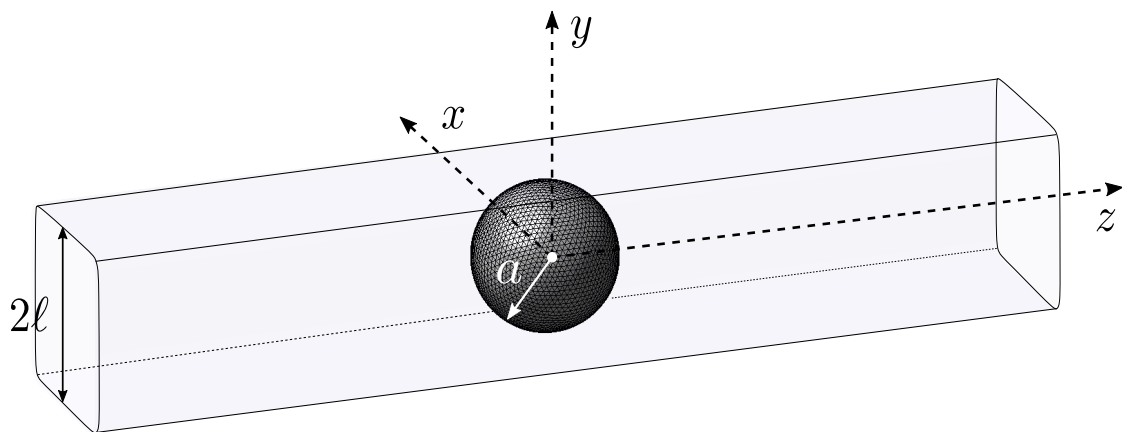
Figure 6. Three-dimensional representation of a capsule flowing in a square microchannel at $T = 0$.

4.2. FOM Result Database Generation

A database of FOM results is generated from a sampling of the parameter domain (see Figure 7). 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 7. 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 $2 \times 3N_x N_c N_t \approx 1.81 \cdot 10^8$ double precision float numbers taking about 1.45 GB of memory.

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 7. 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.
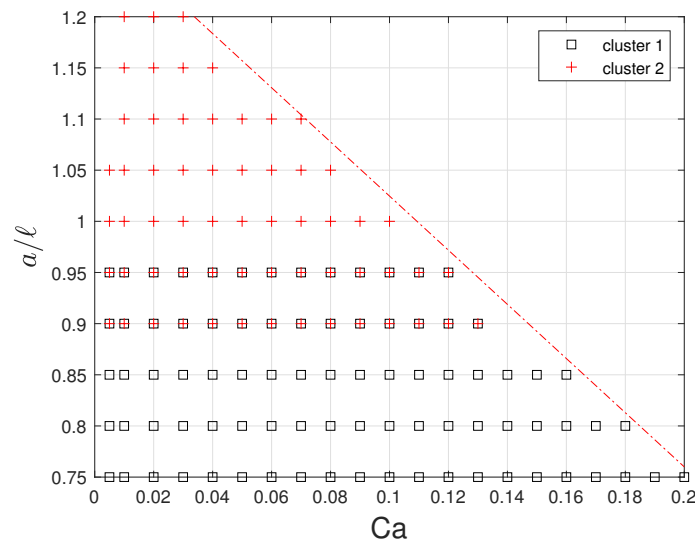
Figure 7. Design of computer experiment with sampling in the admissible parameter domain. The parameter domain is splitted up into two overlapping clusters: cluster 1 (squares), cluster 2 (crosses) and overlapping region (mixed squares and crosses).

4.3. 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 [9]), which is computed as:

$$
\text{RIC}(K) = \frac{\displaystyle\sum_{k=K+1}^{r} \tilde{\sigma}_k^2}{\displaystyle\sum_{k=1}^{r} \tilde{\sigma}_k^2}, \quad K = 1, \ldots, r,
\tag{50}
$$

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

$$
K \mapsto \frac{\tilde{\sigma}_K}{\tilde{\sigma}_1}
\tag{51}
$$

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} [44] 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

$$
d_{MH}(\mathcal{F}, \mathcal{G}) = \max(d_h(\mathcal{F}, \mathcal{G}), d_h(\mathcal{F}, \mathcal{G})),
\tag{52}
$$

with

$$
d_h(\mathcal{F}, \mathcal{G}) = \frac{1}{N_{\mathcal{F}}} \sum_{p_{\mathcal{F}} \in \mathcal{F}} d_s(p_{\mathcal{F}}, \mathcal{G})
\tag{53}
$$

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

$$
d_s(p_{\mathcal{F}}, \mathcal{G}) = \min_{p_{\mathcal{G}} \in \mathcal{G}} \|p_{\mathcal{F}} - p_{\mathcal{G}}\|.
\tag{54}
$$

4.4. 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 8a, we plot the indicator $(1 - \text{RIC})$ (see (50)), 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 8b, 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).
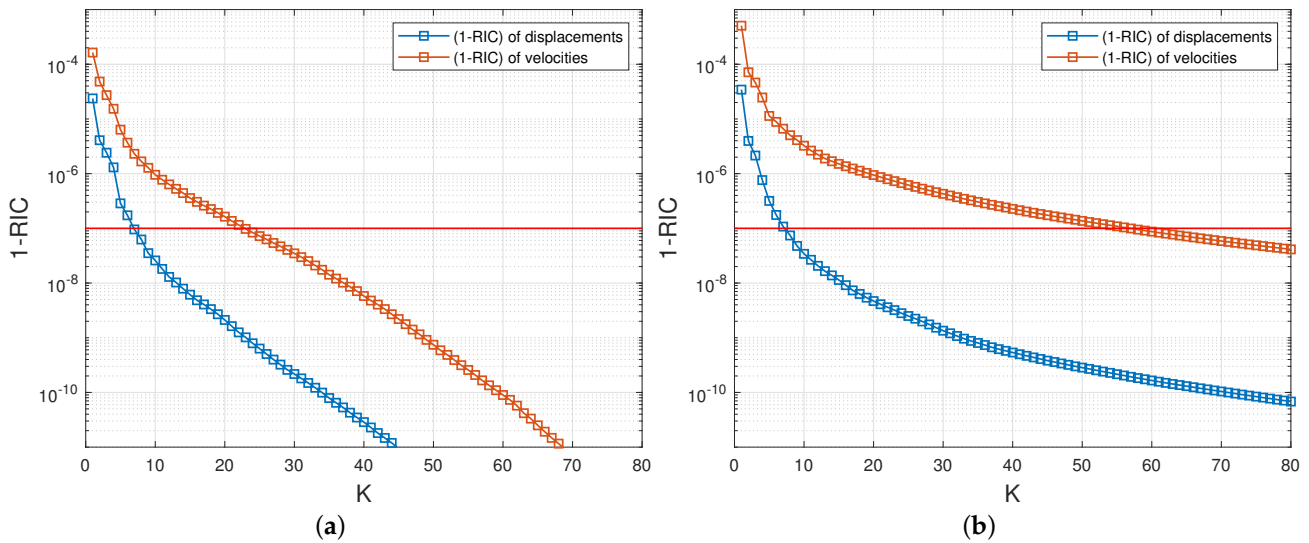
Figure 8. Behaviour of the relative information content of the matrices $\mathcal{S}_u$ and $\mathcal{S}_v$ (**a**) and $\mathcal{T}_u$ and $\mathcal{T}_v$ (**b**) shown in the form $(1 - \text{RIC})$ as a function of the truncation rank K. The horizontal red line corresponds to $(1 - \text{RIC}) = 10^{-7}$.

As supplementary indicators, the singular values $\tilde{\sigma}_K$ normalized by $\tilde{\sigma}_1$ are plotted in Figure 9a (resp. Figure 9b) 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.

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 10). and Figure 11 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).
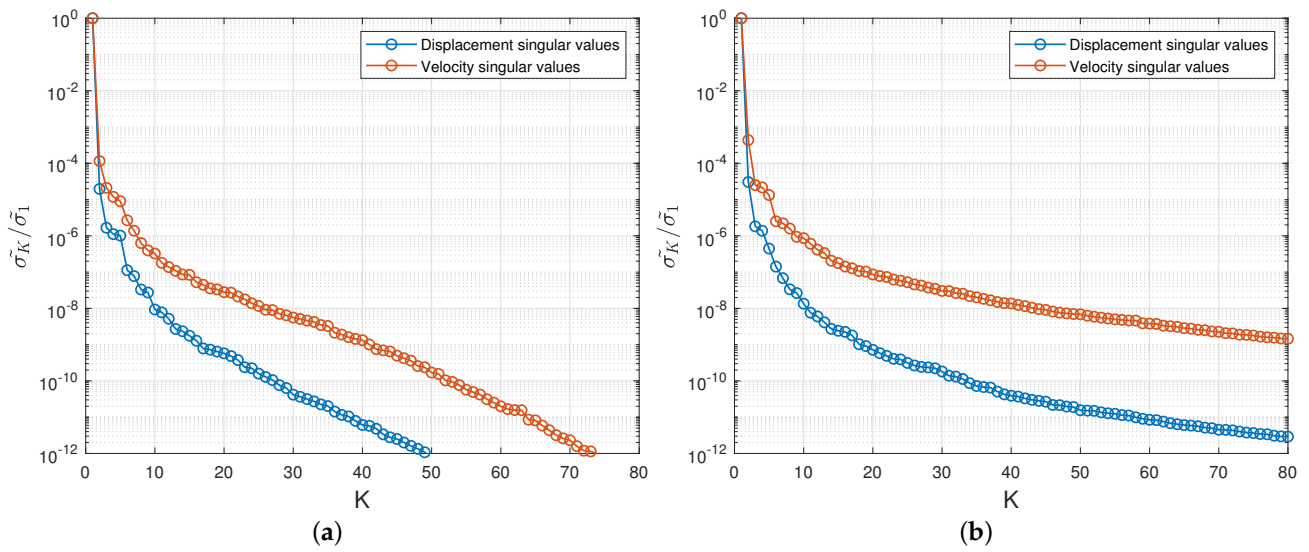
Figure 9. (**a**) Parametric normalized singular values $\tilde{\sigma}_K/\tilde{\sigma}_1$ for $\mathcal{S}_u$ and $\mathcal{S}_v$; (**b**) Spatial normalized singular values $\tilde{\sigma}_K/\tilde{\sigma}_1$ for $\mathcal{T}_u$ and $\mathcal{T}_v$.
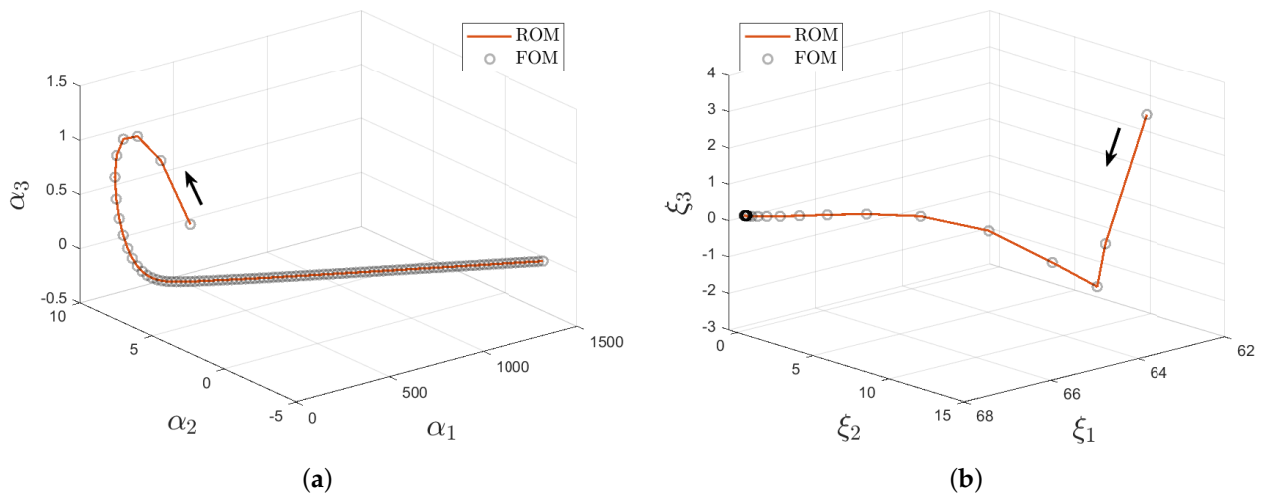

Figure 10. FOM versus ROM comparison of the time evolution of the first three displacement (**a**) and velocity (**b**) POD coefficients for the query parameter $\boldsymbol{\theta}_q = (0.10, 0.90)$.
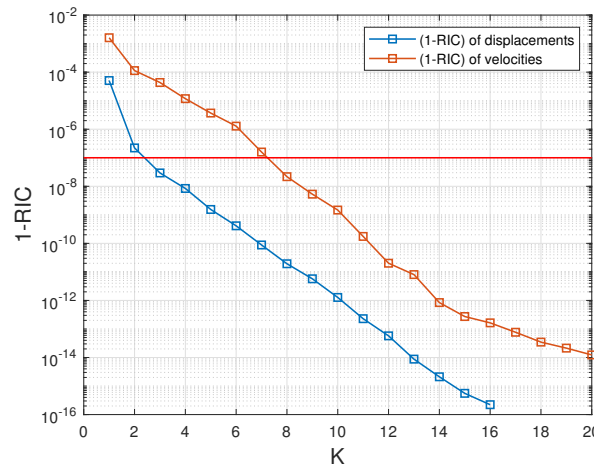

Figure 11. Online stage: behaviour of the relative information content of the matrices $\mathcal{U}(\boldsymbol{\theta}_q)$ and $\mathcal{V}(\boldsymbol{\theta}_q)$ shown in the form $(1 - \text{RIC})$ for query parameter $\boldsymbol{\theta}_q = (0.10, 0.90)$. The red line corresponds to $(1 - \text{RIC}) = 10^{-7}$.

4.5. 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 (25), (30): local second order polynomial reconstruction, $M = 12$;
- For local POD modes: $m_u = 10$, $m_v = 10$;
- For DA in (45), (46): 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 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 $\dot{\gamma}t = 3, 6, 9$.
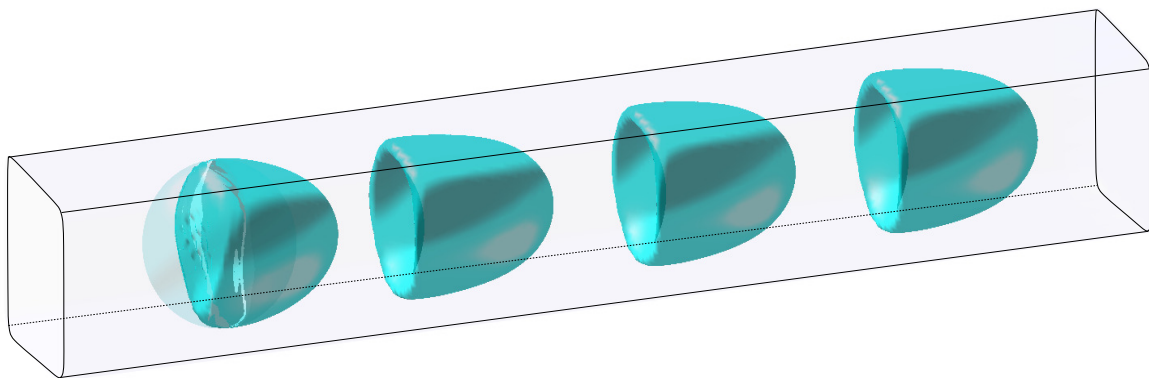
Figure 12. Three-dimensional shape of a capsule flowing in a square microchannel, reconstructed with the ROM model for $\boldsymbol{\theta} = (Ca = 0.10, a/\ell = 0.90)$ and shown at $\dot{\gamma}t = 0, 0.4, 3, 6, 9$. The capsule initial shape is shown in transparency.

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 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 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.
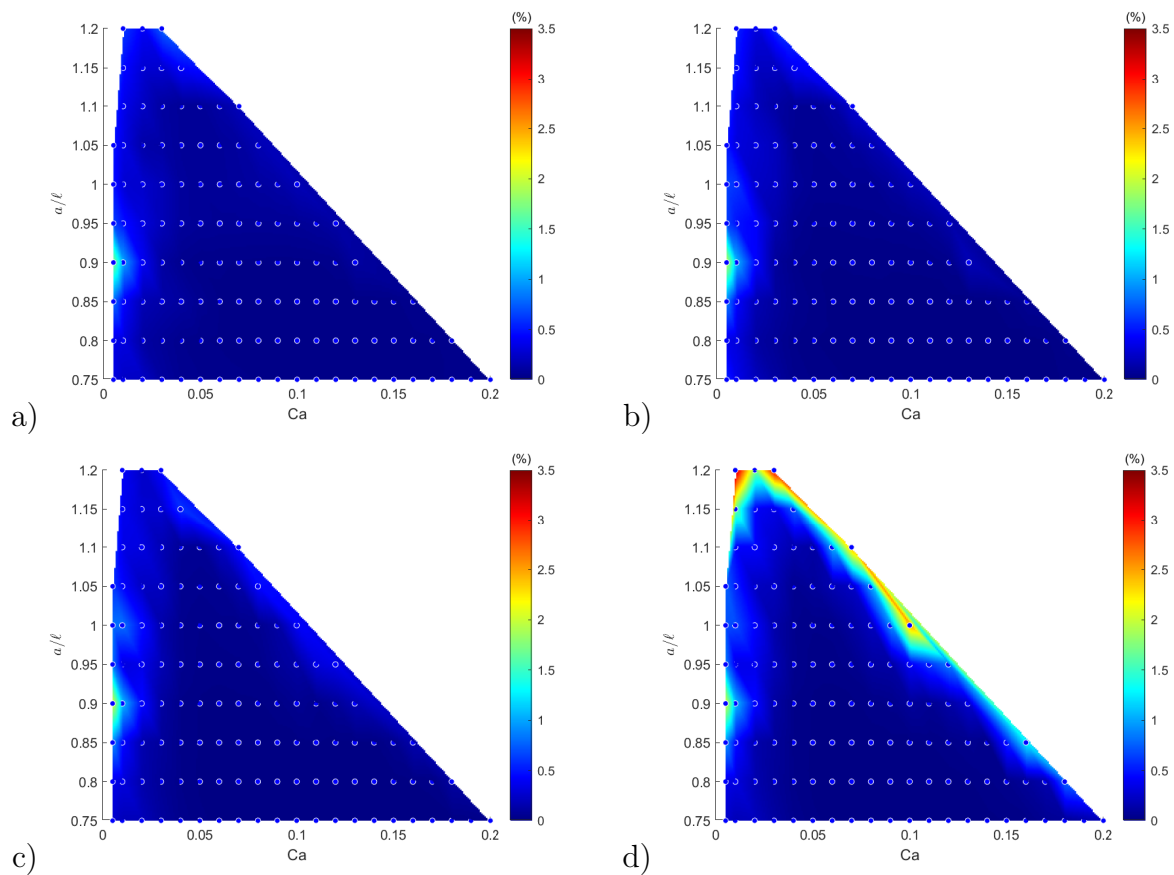
Figure 13. Heat maps of the normalized Hausdorff Distance d_{MH}/a of configuration prediction shapes over the parametric space at different transient states: (**a**) $\dot{\gamma}t = 1$; (**b**) $\dot{\gamma}t = 2$; (**c**) $\dot{\gamma}t = 4$; and (**d**) $\dot{\gamma}t = 8$. The maximum error is 3.26% in (**d**).

The capsule cross-section profiles predicted by the ROM (red dots) are compared to the reference FOM solutions (solid black line) in Figure 14 at different time instants ($\dot{\gamma}t = 0$, 1, 2 and 8) for the 6 configurations, selected as illustration in Figure 3. We observe that the reduced-order model returns very accurate solutions in terms of capsule shape as well-as axial position in the channel.

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.

4.6. CapsuleExplorer: Capsule Visualization/Exploration Software

We have developed an in-house software tool `CapsuleExplorer` based on the proposed ROM to provide the three-dimensional microcapsule deformation/evolution at any time $\dot{\gamma}t$ and for any $\boldsymbol{\theta}_q$ in the admissible parameter domain. `CapsuleExplorer` allows one to select a particular couple $(Ca, a/\ell)$ in the admissible parameter domain, then to visualize the capsule dynamics between initial and final times, either in three dimensions or two dimensions with longitudinal or transversal cross-sectional view. The ROM high performance feature allows real-time exploration/visualization. `CapsuleExplorer` has been developed as a web application. Figures 15 and 16 show some screenshots of the graphics user interface, which will be useful for applications such as identifying the capsule wall mechanical properties through comparison with experimental results.
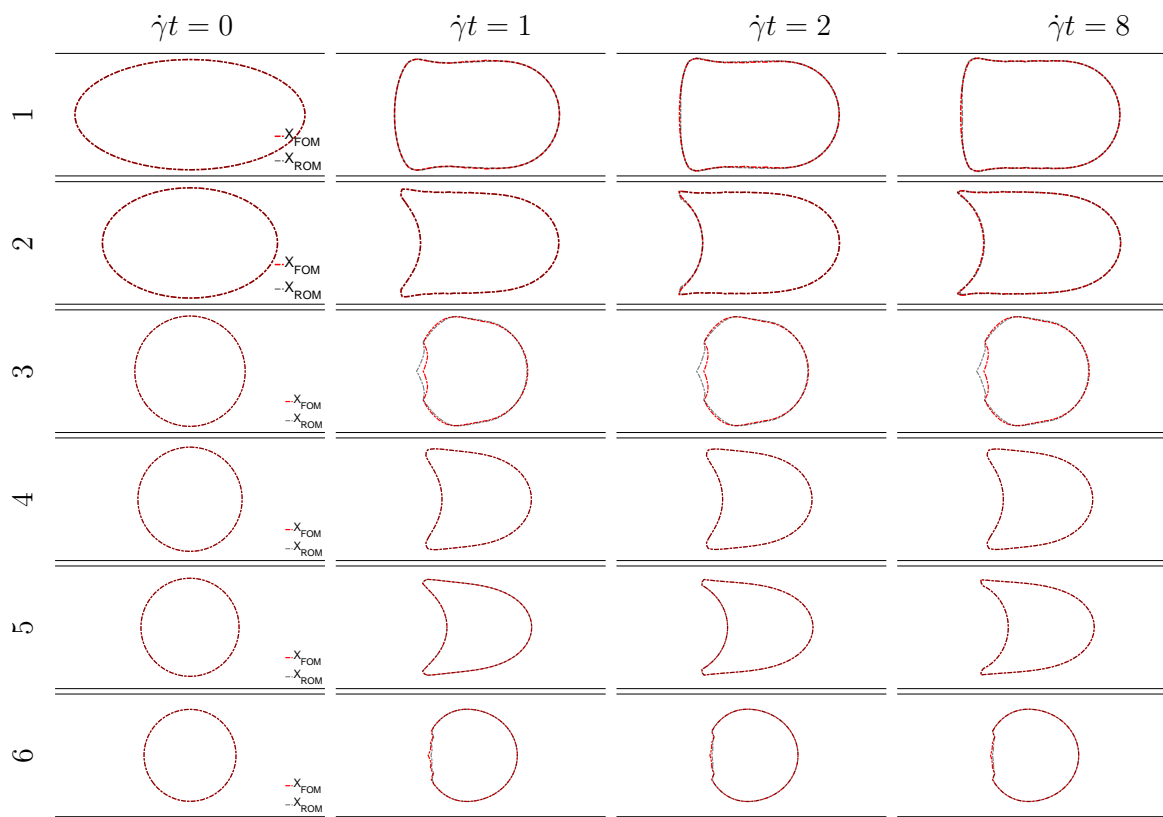
Figure 14. Comparison between the ROM (red dots) and FOM solutions (black line) of the capsule cross-section shapes in the plane $y = 0$ at the times $\dot{\gamma}t = 0$, 1, 2 and 8 respectively, for the 6 parameter couples selected in Figure 3. The horizontal lines correspond to the channel walls.
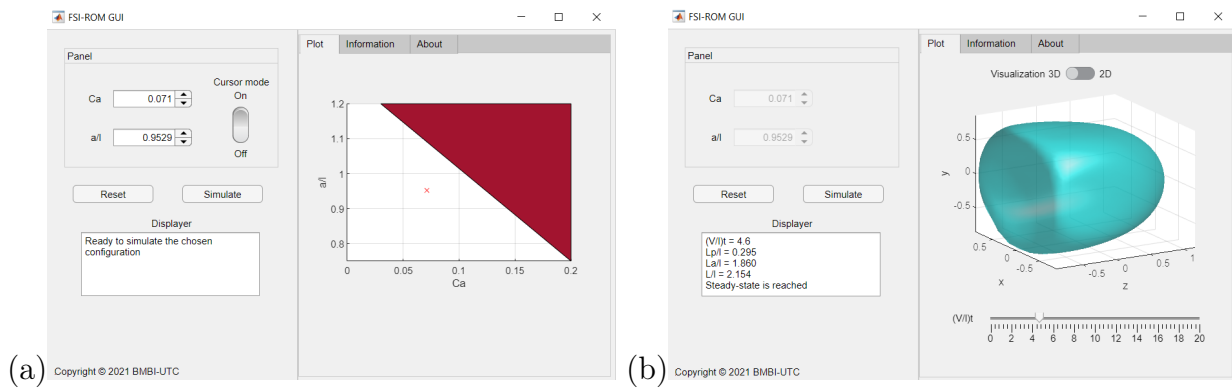

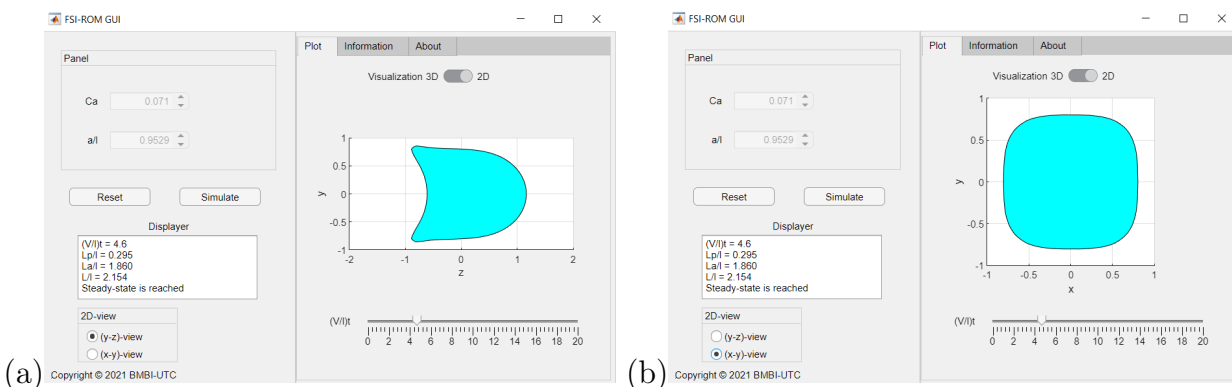**Figure 15.** CapsuleExplorer: (**a**) parameter domain exploration; (**b**) dynamic 3D capsule view.

Figure 16. CapsuleExplorer: (**a**) dynamic 2D cross-section longitudinal view; (**b**) dynamic 2D cross-section transversal view.

5. Concluding Remarks

In this paper, we have presented an innovative data-driven reduced-order model that enables the dynamics of a deformable membrane flowing in a microchannel, from its initial state to the steady shape state. The ROM is built to be valid in a large domain of interest in the parameter plane $(Ca, a/\ell)$. Our FSI-ROM model first starts with an offline procedure to build two global orthonormal bases (space+parameter) that return good approximations of the FOM solutions over the whole parameter domain. The rather small truncation ranks already lead to an appreciable data dimensionality reduction, which is important for complexity and memory storage purposes.

The online stage consists in predicting the space-time solution for any query couple $\boldsymbol{\theta}_q = (Ca, a/\ell)$ in the parameter domain. In a first step, we determine a low-order basis for both the displacement and velocity vector variables. This is achieved by the use of diffuse approximation that returns an interpolated space-time solution at the query vector $\boldsymbol{\theta}_q$. Then an SVD analysis provides a suitable low-order spatial basis for final construction of the ROM in the second step. The physically-based ROM is made of the kinematics equation and the law of membrane quasi-static equilibrium in their reduced formulation. The unknown variables become the POD coefficient vectors of displacement and velocity fields. The reduced quasi-static equilibrium law is determined once again by the use of a diffuse approximation. The manifold learning is achieved by the use of time-snapshot data of the interpolated solution at $\boldsymbol{\theta} = \boldsymbol{\theta}_q$.

Numerical experiments confirm the efficiency of the method. ROM-vs-FOM speedups are observed to be of order 10,000 with almost the same accuracy (with less than a 0.3% error measured in terms of Hausdorff distance inside the parameter domain). Larger errors are encountered at the boundary of the parameter domain, but they still remain reasonable (up to 3.3% in Hausdorff distance). This work tends to show that model-order reduction techniques are complementary and valuable tools for the rapid design and optimization of capsules in healthcare engineering such as drug delivery through blood vessels.

The case of more complex FSI configurations such as the deformation of capsules flowing through a bifurcated microchannel will be investigated in a future work.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Williams, R.J.; Zipser, D. A learning algorithm for continually running fully Recurrent Neural Networks. *Neural Comput.* **1989**, *1*, 270–280. [CrossRef]
2. Raissi, M.; Perdikaris, P.; Karniadakis, G. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *J. Comput. Phys.* **2019**, *378*, 686–707. [CrossRef]
3. Peherstorfer, B.; Willcox, K. Dynamic data-driven reduced-order models. *Comput. Methods Appl. Mech. Eng.* **2015**, *291*, 21–41. [CrossRef]
4. Peherstorfer, B.; Willcox, K. Data-driven operator inference for nonintrusive projection-based model reduction. *Comput. Methods Appl. Mech. Eng.* **2016**, *306*, 196–215. [CrossRef]
5. Pawar, S.; Rahman, S.M.; Vaddireddy, H.; San, O.; Rasheed, A.; Vedula, P. A deep learning enabler for nonintrusive reduced order modeling of fluid flows. *Phys. Fluids* **2019**, *31*, 085101. [CrossRef]
6. Xiao, D.; Fang, F.; Pain, C.; Navon, I. A parameterized non-intrusive reduced order model and error analysis for general time-dependent nonlinear partial differential equations and its applications. *Comput. Methods Appl. Mech. Eng.* **2017**, *317*, 868–889. [CrossRef]

7.　Cordier, L. Proper Orthogonal Decomposition: An overview. In *Lecture Series 2008-01 on Post-Processing of Experimental and Numerical Data*; Von Karman Institute for Fluid Dynamics: Sint-Genesius-Rode, Belgium, 2008

8.　Benner, P.; Gugercin, S.; Willcox, K. A Survey of Projection-Based Model Reduction Methods for Parametric Dynamical Systems. *SIAM Rev.* **2015**, *57*, 483–531. [CrossRef]

9.　Silva, D.F.; Alvaro, L.C. Practical implementation aspects of Galerkin reduced order models based on Proper Orthogonal Decomposition for Computational Fluid Dynamics. *J. Bras. Soc. Mech. Sci. Eng.* **2015**, *37*, 1309–1327. [CrossRef]

10.　Gallivan, K.; Vandendorpe, A.; Van Dooren, P. Model Reduction of MIMO Systems via Tangential Interpolation. *SIAM J. Matrix Anal. Appl.* **2004**, *26*, 328–349. [CrossRef]

11.　Chinesta, F.; Ladeveze, P.; Cueto, E. A short review on model order reduction based on Proper Generalized Decomposition. *Arch. Comput. Methods Eng.* **2011**, *18*, 395. [CrossRef]

12.　Chinesta, F.; Ammar, A.; Cueto, E. On the Use of Proper Generalized Decompositions for Solving the Multidimensional Chemical Master Equation. *Rev. Européenne Mécanique Numérique/Eur. J. Comput. Mech.* **2010**, *19*, 53–64. [CrossRef]

13.　Ghnatios, C.; Masson, F.; Huerta, A.; Leygue, A.; Cueto, E.; Chinesta, F. Proper Generalized Decomposition based dynamic data-driven control of thermal processes. *Comput. Methods Appl. Mech. Eng.* **2012**, *213–216*, 29–41. [CrossRef]

14.　Barrault, M.; Maday, Y.; Nguyen, N.C.; Patera, A.T. An 'empirical interpolation' method: Application to efficient reduced-basis discretization of partial differential equations. *Comptes Rendus Math.* **2004**, *339*, 667–672. [CrossRef]

15.　Chaturantabut, S.; Sorensen, D. Nonlinear model reduction via discrete empirical interpolation. *SIAM J. Sci. Comp.* **2010**, *32*, 2737–2764. [CrossRef]

16.　Xiao, D.; Fang, F.; Buchan, A.; Pain, C.; Navon, O.; Du, J.; Hu, G. Non-linear model reduction for the Navier-Stokes equations using residual DEIM method. *J. Comp. Phys.* **2014**, *263*, 1–18. [CrossRef]

17.　Lappano, E.; Naets, F.; Desmet, W.; Mundo, D.; Nijman, E. 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, Leuven, Belgium, 19–21 September 2016; pp. 3563–3571.

18.　Breitkopf, P.; Rassineux, A.; Villon, P. An Introduction to Moving Least Squares Meshfree Methods. *Rev. Européenne Eléments Finis* **2002**, *11*, 825–867. [CrossRef]

19.　Amsallem, D.; Cortial, J.; Farhat, C. Toward real-time computational-fluid-dynamics-based aeroelastic computations using a database of reduced-order information. *AIAA J.* **2010**, *48*, 2029–2037. [CrossRef]

20.　Audouze, C.; De Vuyst, F.; Nair, P.B. Reduced-order modeling of parameterized PDEs using time–space-parameter principal component analysis. *Int. J. Numer. Methods Eng.* **2009**, *80*, 1025–1057. [CrossRef]

21.　Del Burgo, L.S.; Compte, M.; Aceves, M.; Hernández, R.M.; Sanz, L.; Álvarez-Vallina, L.; Pedraz, J.L. Microencapsulation of therapeutic bispecific antibodies producing cells: Immunotherapeutic organoids for cancer management. *J. Drug Target.* **2015**, *23*, 170–179.

22.　Rabanel, J.M.; Banquy, X.; Zouaoui, H.; Mokhtar, M.; Hildgen, P. Progress technology in microencapsulation methods for cell therapy. *Biotechnol. Prog.* **2009**, *25*, 946–963. [CrossRef]

23.　Chu, T.; Salsac, A.V.; Leclerc, E.; Barthès-Biesel, D.; Wurtz, H.; Edwards-Lévy, F. Comparison between measurements of elasticity and free amino group content of ovalbumin microcapsule membranes: Discrimination of the cross-linking degree. *J. Colloid Interface Sci.* **2011**, *355*, 81–88. [CrossRef]

24.　Hu, X.Q.; Sévénié, B.; Salsac, A.V.; Leclerc, E.; Barthès-Biesel, D. Characterizing the membrane properties of capsules flowing in a square-section microfluidic channel: Effects of the membrane constitutive law. *Phys. Rev. E* **2013**, *87*, 063008. [CrossRef]

25.　de Loubens, C.; Deschamps, J.; Georgelin, M.; Charrier, A.; Edwards-Lévy, F.; Leonetti, M. Mechanical characterization of cross-linked serum albumin microcapsules. *Soft Matter* **2014**, *10*, 4561–4568. [CrossRef] [PubMed]

26.　Sévénié, B.; Salsac, A.V.; Barthès-Biesel, D. Characterization of Capsule Membrane Properties using a Microfluidic Photolithographied Channel: Consequences of Tube Non-squareness. *Procedia IUTAM* **2015**, *16*, 106–114. [CrossRef]

27.　Gubspun, J.; Gires, P.Y.; Loubens, C.d.; Barthès-Biesel, D.; Deschamps, J.; Georgelin, M.; Leonetti, M.; Leclerc, E.; Edwards-Lévy, F.; Salsac, A.V. Characterization of the mechanical properties of cross-linked serum albumin microcapsules: Effect of size and protein concentration. *Colloid Polym. Sci.* **2016**, *294*, 1381–1389. [CrossRef]

28.　De Loubens, C.; Deschamps, J.; Edwards-Lévy, F.; Leonetti, M.; Raissi, M.; Perdikaris, P.; Karniadakis, G.E. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *J. Fluid Mech.* **2016**, *378*, 686–707.

29.　Quesada, C.; Dupont, C.; Villon, P.; Salsac, A.V. Diffuse approximation for identification of the mechanical properties of microcapsules. *Math. Mech. Solids* **2021**, *26*, 1018–1028. [CrossRef]

30.　Wang, Z.; Sui, Y.; Salsac, A.V.; Barthès-Biesel, D.; Wang, W. Motion of a spherical capsule in branched tube flow with finite inertia. *J. Fluid Mech.* **2016**, *806*, 603–626. [CrossRef]

31.　Vesperini, D.; Chaput, O.; Munier, N.; Maire, P.; Edwards-Lévy, F.; Salsac, A.V.; Le Goff, A. Deformability- and size-based microcapsule sorting. *Med. Eng. Phys.* **2017**, *48*, 68–74. [CrossRef] [PubMed]

32.　Wang, Z.; Sui, Y.; Salsac, A.V.; Barthès-Biesel, D.; Wang, W. Path selection of a spherical capsule in a microfluidic branched channel: Towards the design of an enrichment device. *J. Fluid Mech.* **2018**, *849*, 136–162. [CrossRef]

33.　Kabacaoğlu, G.; Biros, G. Sorting same-size red blood cells in deep deterministic lateral displacement devices. *J. Fluid Mech.* **2019**, *859*, 433–475. [CrossRef]

34. Häner, E.; Vesperini, D.; Salsac, A.V.; Le Goff, A.; Juel, A. Sorting of capsules according to their stiffness: From principle to application. *Soft Matter* **2021**, *17*, 3722–3732. [CrossRef]
35. Quesada, C.; Villon, P.; Salsac, A.V. Real-time prediction of the deformation of microcapsules using Proper Orthogonal Decomposition. *J. Fluids Struct.* **2021**, *101*, 103193. [CrossRef]
36. Savignat, J.M. Approximation Diffuse Hermite et Ses Applications. Ph.D. Thesis, École Nationale Supérieure des Mines de Paris, Paris, France, 2000.
37. Raghavan, B.; Breitkopf, P.; Tourbier, Y.; Villon, P. Towards a space reduction approach for efficient structural shape optimization. *Struct. Multidiscip. Optim.* **2013**, *48*, 987–1000. [CrossRef]
38. Hu, X.Q.; Salsac, A.V.; Barthès-Biesel, D. Flow of a spherical capsule in a pore with circular or square cross-section. *J. Fluid Mech.* **2012**, *705*, 176–194. [CrossRef]
39. Wang, X.; Merlo, A.; Dupont, D.; Salsac, A.V.; Barthès-Biesel, D. Characterization of the mechanical properties of microcapsules with a reticulated membrane: Comparison of microfluidic and microrheometric approaches. *Flow* **2021**, in press.
40. Walter, J.; Salsac, A.V.; Barthès-Biesel, D.; Le Tallec, P. Coupling of finite element and boundary integral methods for a capsule in a Stokes flow: Numerical Methods for A Capsule in A Stokes Flow. *Int. J. Numer. Methods Eng.* **2010**, *83*, 829–850. [CrossRef]
41. Barthès-Biesel, D. Modeling the motion of capsules in flow. *Curr. Opin. Colloid Interface Sci.* **2011**, *16*, 3–12. [CrossRef]
42. Golub, G.H.; Reinsch, C. Singular value decomposition and least squares solutions. In *Linear Algebra*; Bauer, F.L., Ed.; Springer: Berlin/Heidelberg, Germany, 1971; pp. 134–151.
43. Buhmann, M.D. *Radial Basis Functions*; Cambridge Monographs on Applied and Computational Mathematics; Cambridge University Press: Cambridge, UK, 2003.
44. Dubuisson, M.P.; Jain, A. A modified Hausdorff distance for object matching. In Proceedings of the 12th International Conference on Pattern Recognition, Jerusalem, Israel, 9–13 October 1994; Volume 1, pp. 566–568.