Space-time-parameter PCA for data-driven modeling with application to Bioengineering

Florian de Vuyst, Claire Dupont, Anne-Virginie Salsac

Chapter 1

Space-time-parameter PCA for data-driven modeling with application to Bioengineering

Florian De Vuyst, Claire Dupont and Anne-Virginie Salsac

Abstract

Principal component analysis is a recognized powerful and practical method in statistics and data science. It can also be used in modeling as a dimensionality reduction tool to achieve low-order models of complex multiphysics or engineering systems. Model-order reduction (MOR) methodologies today are an important topic for engineering design and analysis. Design space exploration or accelerated numerical optimization for example are made easier by the use of reduced-order models. In this chapter, we will talk about the use of higher-order singular value decompositions (HOSVD) applied to spatiotemporal problems that are parameterized by a set of design variables or physical parameters. Here we consider a data-driven reduced order modeling based on a design of computer experiment: from high-dimensional computational results returned by high-fidelity solvers (e.g. finite element ones), the HOSVD allows us to determine spatial, time and parameters principal components. The dynamics of the system can then be retrieved by identifying the low-order discrete dynamical system. As application, we will consider the dynamics of deformable capsules flowing into microchannels. The study of such fluid-structure interaction problems is motivated by the use of microcapsules as innovative drug delivery carriers through blood vessels.

Keywords: HOSVD, spatio-temporal parametrized problem, approximation, reduced-order model, bioengineering, fluid-structure interaction, deformable capsules, dynamical system, machine learning, artificial intelligence

1. Introduction

Manufactured deformable microcapsules are intended to be used as drug carriers within the human vascular network to deliver drugs at specific targets (tumors, etc.). In order to design reliable capsules, one can make help of numerical simulation and high performance computing. The transportation of such capsules into microchannels is a three-dimensional fluid-structure interaction (FSI) problem involving a fluid flow within a confined environment and the deformation of hyperelastic membranes [1, 2]. The behavior of the membrane depends on dimensionless parameters such as the capillary number denoted by Ca and the aspect ratio a/ℓ between the capsule radius a and the channel characteristic length ℓ. The parameter vector $\boldsymbol{\mu} = (Ca, a/\ell)$, for which a capsule

steady shape exists, lies in a bounded domain $D \subset \mathbb{R}^2$. We look for the time evolution of the capsule shape under a Lagrangian description. From an initial shape $\boldsymbol{X}$, we are interested in determining the capsule position $\boldsymbol{x}(\boldsymbol{X}, t, \boldsymbol{\mu})$ in the microchannel domain $\Omega \in \mathbb{R}^3$ at time t for a parameter vector $\boldsymbol{\mu} \in D$. By denoting $\boldsymbol{u}$ the displacement vector from the initial position, we have

$$\boldsymbol{x}(\boldsymbol{X}, \boldsymbol{\mu}, t) = \boldsymbol{X} + \boldsymbol{u}(\boldsymbol{X}, \boldsymbol{\mu}, t) \tag{1}$$

with $\boldsymbol{u}(\boldsymbol{X}, \boldsymbol{\mu}, 0) = \boldsymbol{0}$. The governing equations of the FSI problem include both kinematics and motion equations. At the membrane, we have equilibrium of the mechanical forces (mechanical equilibrium of the membrane and viscous stresses from the fluid). By denoting $\boldsymbol{v}$ the vector field of velocity at the membrane, the system of differential algebraic equations in abstract form reads

$$\dot{\boldsymbol{u}} = \boldsymbol{v}, \tag{2}$$

$$\boldsymbol{v} = \mathcal{F}_{\mu}(\boldsymbol{u}, \boldsymbol{X}). \tag{3}$$

Practically, there are different candidate computational approaches to discretize this system of equations. First, the initial capsule membrane has to be discretized by using a finite element triangular mesh made of nodes $\{\boldsymbol{X}_i\}_{i=1}^{I}$. Regarding time discretization, in [2], an explicit time scheme is used and the velocity field is determined by the use of a boundary integral method (BIM) coupled with a finite element method (FEM). A numerical stability condition imposes the use of small time steps. For a given parameter $\boldsymbol{\mu}$, the time evolution of capsule dynamics on the time intervals of interest generally requires hours of CPU time. To better understand the membrane behavior with respect to $\boldsymbol{\mu}$, a design of computer experiment (DoCE) is done: from a set of J parameter samples of $\boldsymbol{\mu}_j \in D, j = 1, ..., J$, a spatio-temporal solution is computed for each $\boldsymbol{\mu}_j$, leading to a database of shape solutions under the form of a third-order tensor

$$\mathcal{T}_x = \left(\boldsymbol{x}(\boldsymbol{X}_i, \boldsymbol{\mu}_j, t^n)\right)_{i=1,...,I,\, j=1,...,J,\, n=0,...,N,} \in \mathbb{R}^{3I \times J \times (N+1)} \tag{4}$$

using a triangular finite element discretization of the membrane, a time discretization $t^n = n\Delta t$ (assuming that the time step is constant) and the parameter samples $\boldsymbol{\mu}_j$. Typically, for practical computations, $I = O(1000)$, $N = O(10000)$ and $J = O(100)$, so that the tensor database becomes rather huge (about $O(10)$ gigabytes). Of course, one can only store the solutions at coarser times steps and reduce N to $O(100)$ but the database remains rather big in this case.

From this data tensor, one can imagine different use cases leading to different tools:

1. Data exploration and knowledge extraction;

2. Real-time rendering of capsule dynamics for better understanding;

3. Data-driven modeling of capsule dynamics in the whole parameter domain.

First and second items can be achieved by means of data dimensionality reduction. This allows for a lower storage of data in memory as well as a lower numerical complexity of processing and manipulation. In this chapter we will consider a higher order Singular Value Decomposition (HOSVD) which

is a generalization of PCA to tensors. The third item involves a model-order reduction (MOR) methodology. From computed data, we would like to derive a lightweight dynamical system that reproduces the data and, even more, that is able to accurately estimate solutions for any parameter value $\boldsymbol{\mu}$. Data-driven model-order reduction first makes use of data-dimensionality reduction by a low-order tensor decomposition of the solutions according to some suitable spatial, temporal and parameter reduced bases (see [3]). In our application, this will give the truncated decomposition (where the solutions are here seen as functions):

$$\tilde{\boldsymbol{x}}(\boldsymbol{X}, \boldsymbol{\mu}, t) = \boldsymbol{X} + \sum_{k=1}^{K} \sum_{\ell=1}^{L} \sum_{m=1}^{M} a_{k\ell m}\, \boldsymbol{\varphi}_k(\boldsymbol{X})\, \psi_\ell(\boldsymbol{\mu})\, \omega_m(t) \tag{5}$$

for some expansion coefficients $a_{k\ell m}$ and some spatial functions $\boldsymbol{\varphi}_k$, parameter functions ψ_ℓ and temporal functions ω_m with $\omega_m(0) = 0$ ensuring $\boldsymbol{x}(\boldsymbol{X}, \boldsymbol{\mu}, 0) = \boldsymbol{X}$. The truncation ranks K, L and M are expected to be small enough ($K \ll I$, $L \ll J$ and $M \ll N$). Discretized shape solutions returned by the full-order computational model (FOM) are stored into a third-order tensor of data $\mathcal{T}_x$. Let $\tilde{\mathcal{T}}_x$ denote the truncated tensor expansion related to (5). It reads:

$$\tilde{\mathcal{T}}_x = A^0 \otimes \boldsymbol{e}^N + \sum_{k=1}^{K} \sum_{\ell=1}^{L} \sum_{m=1}^{M} a_{k\ell m}\, \boldsymbol{\Phi}_k \otimes \boldsymbol{\Psi}_\ell \otimes \boldsymbol{w}_m \approx \mathcal{T}_x \tag{6}$$

where $A^0 \in \mathbb{R}^{I \times J}$, $(A^0)_{ij} = \boldsymbol{X}_i$, $\boldsymbol{e}^N = (1, ..., 1)^T \in \mathbb{R}^N$, $\boldsymbol{\Phi}_k \in \mathbb{R}^I$, $\boldsymbol{\Psi}_\ell \in \mathbb{R}^J$ and $\boldsymbol{w}_m \in \mathbb{R}^N$. In this chapter, we will seamlessly use the functional representation or the tensor one.

From this reduced form, we will then apply a kernel-based Dynamic Mode Decomposition (k-DMD, see [4]) to derive a dynamical system able to predict the capsule shape evolution over time for any parameter $\boldsymbol{\mu}$. This will be developed in the next sections.

2. Higher-order singular value decomposition and truncation

2.1 Compact HOSVD

The higher-order singular value decomposition (HOSVD) of a tensor is a specific orthogonal Tucker decomposition. The classical computation of a HOSVD was introduced by L. R. Tucker [5] and further developed by L. De Lathauwer et al. [6]. Robust computations or improvements have been since proposed [6–8]. For a tensor $\mathcal{T}$ of order d, the idea is to compute the singular value decomposition of each factor-k flattening $T_{(k)}$ of a tensor $\mathcal{T}$, i.e. a "matricisation" of the tensor where the rows of the matrix are related to the k-th dimension.

In our case, we consider a third-order tensor and successive spatial, parameter and temporal flattening to find the singular vectors. The spatial tensor flattening $T_{(X)} \in \mathbb{R}^{I \times (J.N)}$ of the tensor $\left(\mathcal{T}_x - A^0 \otimes \boldsymbol{e}^N\right)$ is

$$\left(T_{(X)}\right)_{ip} = \boldsymbol{x}(\boldsymbol{X}_i, \boldsymbol{\mu}_j, t^n) - \boldsymbol{X}_i, \qquad p = j + (n-1)J$$

(meaning that the $\boldsymbol{\mu}$ and t dimensions are stacked in columns in the matrix). The SVD of $T_{(X)}$ provides r_x nonzero singular values with r_x corresponding singular orthonormal vectors $\boldsymbol{\Phi}_k$, $k = 1, ..., r_x$. Similarly the parameter flattening

leads to a rank r_μ with r_μ modes $\mathbf{\Psi}_\ell$, $\ell = 1, ..., r_\mu$ and the time flattening a rank r_t with r_t modes $\boldsymbol{w}_m$, $m = 1, ..., r_t$. The tuple (r_X, r_μ, r_t) is the multilinear rank of $\mathcal{T}_x$. Then tensor $\mathcal{T}_x$ can be written as

$$\mathcal{T}_x = A^0 \otimes \boldsymbol{e}^N + \sum_{k=1}^{r_X} \sum_{\ell=1}^{r_\mu} \sum_{m=1}^{r_t} a_{k\ell m}\, \mathbf{\Phi}_k \otimes \mathbf{\Psi}_\ell \otimes \boldsymbol{w}_m. \tag{7}$$

2.2 Approximation

Among the applications, HOSVD can be used to define a low-order approximation of tensors. The so-called truncated HOSVD [9–11] consists in truncating the expansion (7) at a given multilinear rank (K, L, M), $K \leq r_X, L \leq r_\mu, M \leq r_t$, leading to (6). Let mlrank$(\mathcal{T})$ denote the multilinear rank of the tensor $\mathcal{T}$. It has been shown that the approximation (6) returns a quasi-optimal solution of the nonlinear non-convex least-square problem

$$\min_{\tilde{\mathcal{T}}} \quad \frac{1}{2} \|\mathcal{T}_x - \tilde{\mathcal{T}}\|_F^2 \tag{8}$$

(here $\|.\|_F$ is the Frobenius norm for tensors) subject to mlrank$(\tilde{\mathcal{T}}) = (K, L, M)$. The truncation ranks can be determined *a priori* according to the classical relative information content (RIC) of the SVD theory.

Eq. (6) can already be used as a compressed representation of the data, allowing for a lower storage complexity and a simpler manipulation, with low information loss if the RIC is high.

3. Reduced-order modeling of capsule dynamics

Eq. (6) provides a summarization of the family of spatio-temporal capsule shape solutions in the time interval $[t^0, t^N]$. Unfortunately, this algebraic model has no predictability capability for time $t > t^N$. To derive a predictable time-dependent model from the data tensor $\mathcal{T}_x$, one has to derive a differential system that approximates the FSI system of equations (2) and (3). The HOSVD reduction can thus be valued in the context of model-order reduction.

Consider of parameter vector of interest $\boldsymbol{\mu} \in D$. The capsule position approximate solution (5) can be rewritten as

$$\tilde{\boldsymbol{x}}(\boldsymbol{X}, \boldsymbol{\mu}, t) = \boldsymbol{X} + \sum_{k=1}^{K} a_{\mu,k}(t)\, \boldsymbol{\varphi}_k(\boldsymbol{X}) \tag{9}$$

with

$$a_{\mu,k}(t) = \sum_{\ell=1}^{L} \sum_{m=1}^{M} a_{k\ell m} \psi_\ell(\boldsymbol{\mu}) \omega_m(t). \tag{10}$$

Let $\boldsymbol{a}_\mu(t)$ be the vector-valued function

$$\boldsymbol{a}_\mu(t) = \big(a_{\mu,1}(t), ..., a_{\mu,K}(t)\big)^T \in \mathbb{R}^K.$$

We would like to derive a differential system of unknowns $\boldsymbol{a}_\mu(t)$. Since $\boldsymbol{x}(\boldsymbol{X}, \boldsymbol{\mu}, 0) = \boldsymbol{X}$, we have the natural initial condition $\boldsymbol{a}_\mu(0) = \boldsymbol{0}$. Practically, from expansion (7), we can compute coefficients $a_{\mu,k}(t)$ at discrete times t^n, and have thus access to the list of coefficient vectors

$$\boldsymbol{a}_\mu^n = \boldsymbol{a}_\mu(t^n), \quad n = 0, ..., N. \tag{11}$$

A dynamical reduced-order model consists in determining (or approximating) a Lipschitz continuous mapping $\boldsymbol{F}_\mu : \mathbb{R}^K \to \mathbb{R}^K$ such that

$$\boldsymbol{a}_\mu^0 = \boldsymbol{0}, \quad \boldsymbol{a}_\mu^{n+1} \approx \boldsymbol{F}_\mu(\boldsymbol{a}_\mu^n) \quad \forall n \in 0, ..., N-1 \tag{12}$$

from the data $\{\boldsymbol{a}_\mu^n\}_{n=0}^N$. We get a low-order discrete dynamical system. Note that, here, we do not search for parameters of a model, but for the equations of the model themselves. Since the problem of finding such a mapping is infinite dimensional, one has to restrict the search to a mapping in a (suitable) finite dimensional functional space.

4. Koopman theory and dynamic mode decomposition

4.1 Koopman operator for discrete dynamical systems

Koopman theory is a powerful mathematical framework that re-expresses a general nonlinear discrete dynamical system as the knowledge of a linear (infinite dimensional) operator, the so-called Koopman operator or compositional operator. Today it is commonly used in machine learning and data-driven model-order reduction methodologies [4, 12]. Let us assume a discrete dynamical system in the form

$$\boldsymbol{a}^{n+1} = \boldsymbol{F}(\boldsymbol{a}^n), \quad n \in \mathbb{N} \tag{13}$$

for a Lipschitz continuous mapping $\boldsymbol{F}$ from $\mathbb{R}^d$ to $\mathbb{R}^d$. Let g be a function of a Banach space X, $g : \mathbb{R}^d \to \mathbb{R}$. So we have $g(\boldsymbol{a}^{n+1}) = (g \circ \boldsymbol{F})(\boldsymbol{a}^n)$. The Koopman operator related to $\boldsymbol{F}$ is defined as

$$\mathcal{K}g = g \circ \boldsymbol{F} \quad \forall g \in X. \tag{14}$$

Then we have

$$g(\boldsymbol{a}^{n+1}) = (\mathcal{K}g)(\boldsymbol{a}^n).$$

The knowledge of $\mathcal{K}$ includes the knowledge of $\boldsymbol{F}$. Indeed, by taking the particular observables $g_i(\boldsymbol{a}) = \boldsymbol{a} \cdot \boldsymbol{e}_i$, $i = 1, ..., d$ where $\boldsymbol{e}_i$ is the i-th vector of the canonical basis of $\mathbb{R}^d$, we retrieve $(\boldsymbol{a}^{n+1})_i = F_i(\boldsymbol{a}^n)$, i.e. the i-th equation of (13). Of course, the linear Koopman operator acts on an infinite-dimensional functional space, so it is impossible to determine it exactly. However, one can search for an approximate Koopman operator $\tilde{\mathcal{K}}$ that acts on an approximate finite-dimensional space $\tilde{X} \subset X$.

The concept of (nonlinear) observables is to have a sufficiently large set of independent nonlinear functions of the state vector and measurements of them in order to identify the mapping $\boldsymbol{F}$. A natural question of interest is what are

the best observables to choose. There is no absolute answer to this question, and the choice may depend on the underlying Physics. Without any a priori knowledge on the system of equations, one can use basis functions of a universal approximators like polynomials, Fourier or kernel-based functions for example.

4.2 Dynamic mode decomposition

The simplest choice of observables is the linear functions $g_i(\boldsymbol{a}) = \boldsymbol{a} \cdot \boldsymbol{e}_i$, $i = 1, ..., d$. It leads to the search of a finite-dimensional approximation A of $\mathcal{T}$ from the full state vector data. The matrix A can be searched as the solution of the least square minimization problem

$$\min_{A \in \mathscr{M}_d(\mathbb{R})} \quad \frac{1}{2}\|Y - AX\|_F^2 \tag{15}$$

where $X = [\boldsymbol{a}^0, \boldsymbol{a}^1, ..., \boldsymbol{a}^{N-1}]$ and $Y = [\boldsymbol{a}^1, \boldsymbol{a}^2, ..., \boldsymbol{a}^N]$. Assuming $N \geq d$, the solution is given by $A = YX^T(XX^T)^{-1}$. The least square problem (15) can be eventually regularized for better conditioning by a Tykhonov regularization term [13, 14]. This practical approach of Koopman operator approximation is referred to as the dynamic mode decomposition (DMD) [4, 12]. This provides a linear dynamical model

$$\boldsymbol{a}^{n+1} = A\boldsymbol{a}^n \tag{16}$$

starting from a given initial condition $\boldsymbol{a}^0$. The solution of (16) which is $\boldsymbol{a}^n = A^n\boldsymbol{a}^0$ is bounded for any initial condition $\boldsymbol{a}^0$ as soon as $\rho(A) \leq 1$.

5. Kernel-based identification of dynamical systems

In the case of a strongly linear dynamical system, the linear model (16) can be not accurate enough. We have to include suitable nonlinear observables in the data and the model. In this section, observables are selected from kernel-based approximations [15]. Then we use the variant kernel-DMD (k-DMD, [4]) approach to identify $\boldsymbol{F}$.

A real-valued function k on $\mathbb{R}^d \times \mathbb{R}^d$ is called a positive definite kernel function if it is symmetric and if the following property holds:

$$\forall m \in \mathbb{N}^\star,\ \forall \{\boldsymbol{z}_i\}_{i=1}^m \in (\mathbb{R}^d)^m, \forall \{\alpha_i\}_{i=1}^m \in \mathbb{R}^m, \quad \sum_{i=1}^m \sum_{j=1}^m \alpha_i\, \alpha_j\, k(\boldsymbol{z}_i, \boldsymbol{z}_j) \geq 0.$$

In other words, the square matrix $\boldsymbol{K} = \big(k(\boldsymbol{z}_i, \boldsymbol{z}_j)\big)_{i=1,...,m,\, j=1,...,m}$ is positive semi-definite. A standard kernel function is for example the Gaussian one

$$k(\boldsymbol{z}, \boldsymbol{z}') = \exp\left(-\frac{1}{2}\frac{\|\boldsymbol{z} - \boldsymbol{z}'\|^2}{\sigma^2}\right) \tag{17}$$

for a given parameter $\sigma > 0$.

5.1 Kernel-based interpolation

Kernel functions can be used for interpolation in spaces of arbitrary dimension. Let $g : \mathbb{R}^d \to \mathbb{R}$ be a continuous function and assume that we know the values of $g(\boldsymbol{z}_i)$ at particular points $\boldsymbol{z}_i$, $i = 1, ..., m$. then one can define an interpolator $\mathcal{I}g$ of g defined as

$$\mathcal{I}g(\boldsymbol{z}) = \sum_{j=1}^{m} \alpha_j \, k(\boldsymbol{z}, \boldsymbol{z}_j) \tag{18}$$

where the coefficient vector $\boldsymbol{\alpha} = (\alpha_1, ..., \alpha_m)^T$ is determined such that the interpolation property

$$\mathcal{I}g(\boldsymbol{z}_i) = g(\boldsymbol{z}_i) \quad \forall i \in \{1, ..., m\}.$$

holds. The interpolation conditions clearly lead to the solution of the symmetric linear square system of size m

$$\boldsymbol{K}\boldsymbol{\alpha} = \boldsymbol{b} \tag{19}$$

where $\boldsymbol{b} = (g(\boldsymbol{z}_1), ..., g(\boldsymbol{z}_m))^T)$. Assuming that $\boldsymbol{K}$ is positive definite, then Eq. (19) has a unique solution. Let

$$k_i(\boldsymbol{z}) = k(\boldsymbol{z}, \boldsymbol{z}_i), \quad i = 1, ..., m$$

and $V = \text{span}(k_1, ..., k_m)$. Considering any function $\tilde{g} \in V$, it is easy to check that $\tilde{g} = \mathcal{I}\tilde{g}$. One can derive the interpolation error

$$\begin{aligned} \|g - \mathcal{I}g\|_{L^\infty} &= \|g - \tilde{g} + \tilde{g} - \mathcal{I}g\|_{L^\infty} \\ &= \|g - \tilde{g} + \mathcal{I}\tilde{g} - \mathcal{I}g\|_{L^\infty} \\ &\leq \|g - \tilde{g}\|_{L^\infty} + |||\mathcal{I}||| \, \|g - \tilde{g}\|_{L^\infty} \end{aligned}$$

so that

$$\|g - \mathcal{I}g\|_{L^\infty} \leq (1 + |||\mathcal{I}|||) \inf_{\tilde{g} \in V} \|g - \tilde{g}\|_{L^\infty}. \tag{20}$$

The interpolation error is controlled by the best approximation error multiplied by a stability constant depending on the norm of the interpolation operator.

5.2 Use of kernel features and k-DMD

Let us go back to the parameterized dynamical system of interest (12) and consider a point cloud $\{\boldsymbol{a}_{(j)}\}_{j=1}^m$ of sample states in the admissible reduced state space $\mathcal{X} \subset \mathbb{R}^K$. The functions

$$k_j(\boldsymbol{a}) = k(\boldsymbol{a}, \boldsymbol{a}_{(j)})$$

can be seen as features and thus be used as suitable nonlinear observables to approximate the Koopman operator. From any known full state vector $\boldsymbol{a}_\mu^n$ at

time t^n, we build the vector of observables

$$\boldsymbol{\kappa}(\boldsymbol{a}_\mu^n) = \left(k_1(\boldsymbol{a}_\mu^n),\, k_2(\boldsymbol{a}_\mu^n), ..., k_m(\boldsymbol{a}_\mu^n)\right)^T. \tag{21}$$

By definition of the Koopman operator, we have $k_i(\boldsymbol{a}_\mu^{n+1}) = (k_i \circ \boldsymbol{F})(\boldsymbol{a}_\mu^n) = (\mathcal{K}k_i)(\boldsymbol{a}_\mu^n)$. Then we look for a finite-dimensional approximation A_μ of the Koopman operator $\mathcal{K}$ in the sense

$$\boldsymbol{\kappa}(\boldsymbol{a}_\mu^{n+1}) \approx A_\mu\, \boldsymbol{\kappa}(\boldsymbol{a}_\mu^n) \quad \forall n \in \{0, ..., N-1\} \tag{22}$$

The matrix A_μ is searched as the minimum of the least square problem

$$\min_{A \in \mathscr{M}_m(\mathbb{R})} \quad \frac{1}{2}\|Y_\mu - A\, X_\mu\|_F^2$$

where the entry and output data matrices are now $X_\mu = [\boldsymbol{\kappa}(\boldsymbol{a}_\mu^0), \boldsymbol{\kappa}(\boldsymbol{a}_\mu^1), ..., \boldsymbol{\kappa}(\boldsymbol{a}_\mu^{N-1})]$ and $Y_\mu = [\boldsymbol{\kappa}(\boldsymbol{a}_\mu^1), \boldsymbol{\kappa}(\boldsymbol{a}_\mu^2), ..., \boldsymbol{\kappa}(\boldsymbol{a}_\mu^N)]$. We get the dynamical system $\boldsymbol{\kappa}(\boldsymbol{a}_\mu^{n+1}) = A_\mu\, \boldsymbol{\kappa}(\boldsymbol{a}_\mu^n)$ with a specified initial condition $\boldsymbol{\kappa}(\boldsymbol{a}_\mu^0)$.

Let us emphasize that the computational variables are now the $\boldsymbol{\kappa}(\boldsymbol{a}_\mu^n)$. But we still need the full state variables $\boldsymbol{a}^n$ to determine the displacements or the capsule shapes. The full state can be retrieved for example by interpolation: taking $g_i(\boldsymbol{a}) = \boldsymbol{a} \cdot \boldsymbol{e}_i$, we get

$$(\boldsymbol{a}_\mu)_i^{n+1} \approx \mathcal{I}g_i(\boldsymbol{a}_\mu^{n+1}) = \sum_{j=1}^{m} (\boldsymbol{\alpha}_j)_i\, k_i(\boldsymbol{a}_\mu^{n+1})$$

with interpolation coefficients vectors $\{\boldsymbol{\alpha}_j\}_{j=1}^m$ such that

$$\boldsymbol{a}_{(i)} = \sum_{j=1}^{m} \boldsymbol{\alpha}_j\, k_j(\boldsymbol{a}_{(i)}) \quad \forall i \in \{1, ..., m\}$$

(linear system of dimension $m \times K$). The coefficient vectors $\boldsymbol{\alpha}_j$ can be computed once for all.

5.3 Including the full state vector and the constants into the features

The k-DMD approach can be improved by including the full-state vector $\boldsymbol{a}_\mu$ itself into the input feature vector. Moreover, if the kernel functions are not able to perfectly reproduce the constant ones, for accuracy reasons it is justified to add the constants into the features. Considering the augmented vector $\boldsymbol{\eta}(\boldsymbol{a}_\mu)$ of observables

$$\boldsymbol{\eta}(\boldsymbol{a}_\mu) = \begin{bmatrix} \boldsymbol{a}_\mu \\ \boldsymbol{\kappa}(\boldsymbol{a}_\mu) \\ 1 \end{bmatrix} \in \mathbb{R}^{K+m+1}, \tag{23}$$

we look for a dynamical system in the form

$$\boldsymbol{a}_\mu^{n+1} = A_\mu\, \boldsymbol{\eta}(\boldsymbol{a}_\mu^n) \approx \boldsymbol{F}_\mu(\boldsymbol{a}_\mu^n) \tag{24}$$

with a constant rectangular matrix A_μ of size $K \times (K + m + 1)$ to identify. The first advantage is that the output vector is the full state itself. The second one is the number of elements of A_μ which is less than m^2 as soon as $K(K + m + 1) \leq m^2$. In this case, there are less coefficients to identify. The input and output matrices now are

$$X_\mu = [\boldsymbol{\eta}(\boldsymbol{a}_\mu^0), \boldsymbol{\eta}(\boldsymbol{a}_\mu^1), ..., \boldsymbol{\eta}(\boldsymbol{a}_\mu^{N-1})], \quad Y_\mu = [\boldsymbol{a}_\mu^1, \boldsymbol{a}_\mu^2, ..., \boldsymbol{a}_\mu^N]. \tag{25}$$

Assuming $N \geq (K + m)$, the rectangular matrix A_μ solution of the least square problem

$$\min_{A \in \mathcal{M}_{K,K+m+1}(\mathbb{R})} \quad \frac{1}{2}\|Y_\mu - A\, X_\mu\|_F^2 \tag{26}$$

is computed as $A_\mu = Y_\mu X_\mu^T (X_\mu X_\mu^T)^{-1}$.

5.4 Summary and whole algorithm

We give a summary of the model-order reduction algorithm:

1. Input data: third-order tensor $\mathcal{T}_x$ (4) made of capsule shape solutions of size $3I \times J \times (N + 1)$.

2. HOSVD + truncated approximation: compute (7) and get the truncated approximation with the truncated multilinear ranks (K, L, M):

$$\tilde{\mathcal{T}}_x = A^0 \otimes \boldsymbol{e}^N + \sum_{k=1}^{K}\sum_{\ell=1}^{L}\sum_{m=1}^{M} a_{k\ell m}\, \boldsymbol{\Phi}_k \otimes \boldsymbol{\Psi}_\ell \otimes \boldsymbol{w}_m$$

3. Online stage: choose a parameter vector $\boldsymbol{\mu}$. From (11), compute the data coefficients $\boldsymbol{a}_\mu(t^n) = \boldsymbol{a}_\mu^n$ (see Eq. (11)) from (10). Choose a kernel function $k(.,.)$, choose m and the centers $\boldsymbol{a}_{(j)}$, $j = 1, ..., m$. Assemble the observables $\boldsymbol{\eta}(\boldsymbol{a}_\mu^n)$ and assemble the matrices X_μ and Y_μ (Eq. (25)). Compute the k-DMD matrix $A_\mu = Y_\mu X_\mu^T (X_\mu X_\mu^T)^{-1}$. We get the reduced-order dynamical system (24) with $\boldsymbol{a}_\mu^0 = \boldsymbol{0}$ as initial value.

6. Numerical results for capsule dynamics

The algorithm is applied to a problem of deformable capsules flowing into a square-based microchannel (typical for microfluidic channels created by soft lithography) with a bulk inflow velocity V [1, 2]. Related works and variant ROM approaches on this topic can be found in [14, 18]. The capsule dynamics is simulated by the full-order fluid-structure interaction solver. The fluid solver is based on the solution of the Stokes equations and a nonlinear Neo-Hookean law is used for the membrane. The initial capsule is spherical, corresponding to the shape at rest. The sphere is discretized with a finite element mesh made of $I = 2562$ vertices.

6.1 Study for a particular parameter couple $(Ca, a/\ell)$

The objective of this subsection being to illustrate the methods on an example and show how to apply them, we only consider the snapshot FOM solutions for $(Ca, a/\ell) = (0.1, 0.9)$ for the sake of simplicity and brevity. Figure 1 shows both membrane shapes and positions in the channel at different instants. At each time

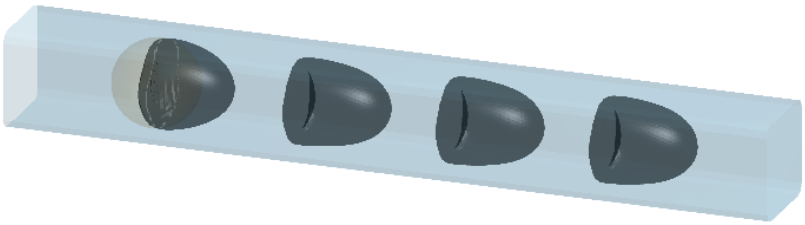

Figure 1.
Example of a microcapsule dynamics within a square-base channel for $(Ca, a/\ell) = (0.1, 0.90)$. Shapes and locations of an initially spherical capsule are shown at $t = 0$ (in transparency), 0.4, 2.8, 5.2, 7.6.

$t^n = n\Delta t$, where the time step is equal to $\Delta t = 0.04$, a snapshot is saved and stored in the database. Note that all time quantities are adimensionalized by the factor ℓ/V. The resulting generated data matrix is used as the entry matrix for the ROM learning process. Then a truncated SVD is applied to get the spatial POD modes. In Figure 2, the four first eigenmodes Φ^k are plotted (more precisely this is a combination of each mode with the original spherical shape for better visualization and understanding of their influence). In Figure 3, the graphics $k \mapsto 1 - RIC(k)$ in log scale is plotted. We decide to use a truncation rank K equal to $K = 10$, returning a relative information content of about $1 - 3.5 \times 10^{-5}$.

Then a reduced-order dynamical system for the capsule time evolution is searched. In this example, we compare two models: the first one is the affine approximation (denoted by ROM-A)

$$\boldsymbol{a}_\mu^{n+1} = A_\mu \boldsymbol{a}_\mu^n + \boldsymbol{b}_\mu$$

with the matrix A and the vector $\boldsymbol{b}$ to identify. It is equivalent to consider the vector of features $\boldsymbol{\eta}(\boldsymbol{a}) = (\boldsymbol{a}, 1)^T$. The second nonlinear model is built from the observable vector $\boldsymbol{\eta}(\boldsymbol{a}) = (\boldsymbol{a}, \boldsymbol{\kappa}(\boldsymbol{a}), 1)^T$ with the recurrent time scheme

$$\boldsymbol{a}_\mu^{n+1} = A_\mu \boldsymbol{\eta}(\boldsymbol{a}_\mu^n)$$

(ROM-B). For ROM-B, the Gaussian kernel function (17) is used. The standard deviation parameter σ in (17) is chosen as $\sigma = \max_n \|\boldsymbol{a}_\mu^n\| = \|\boldsymbol{a}_\mu^{N+1}\|_2$. In both cases ROM-A and ROM-B, the determination of the matrix A_μ by minimization of the least square problem (26) leads to a very small residual. In Figure 4, the logarithm of the relative time residual

$$n \mapsto \log_{10}\left(\|\boldsymbol{a}_\mu^{n+1} - A_\mu \boldsymbol{\eta}(\boldsymbol{a}_\mu^n)\|^2 / \|\boldsymbol{a}_\mu^{n+1}\|^2\right)$$

is plotted for each ROM model. One can observe values between 10^{-14} and 10^{-8}. The residual for ROM-B appears to be slightly smaller than that of ROM-A

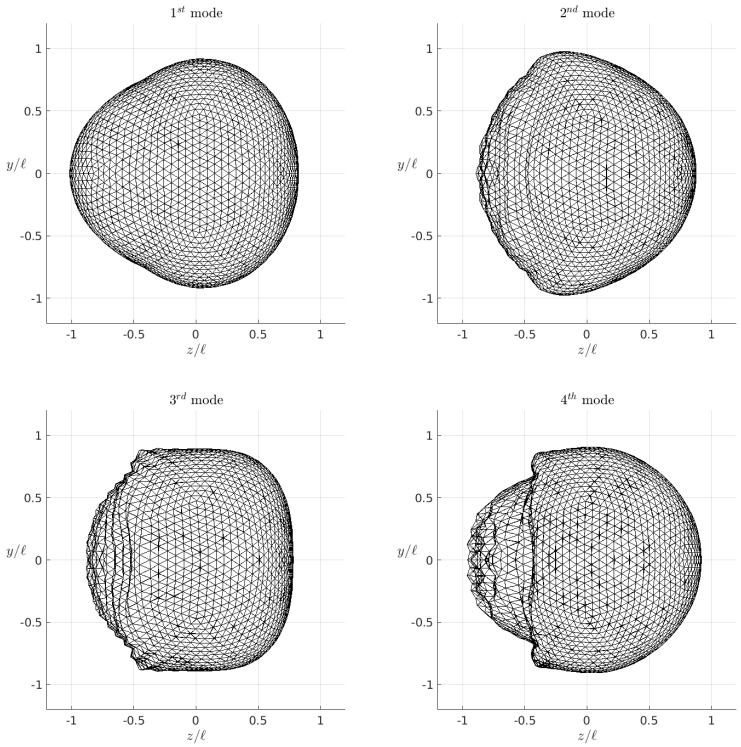

Figure 2.
SVD: four first spatial principal components computed by the HOSVD. Each mode has been added on the initial spherical shape and amplified by a factor 2 for better visualization. Higher-order modes show oscillations at the rear of the capsule.

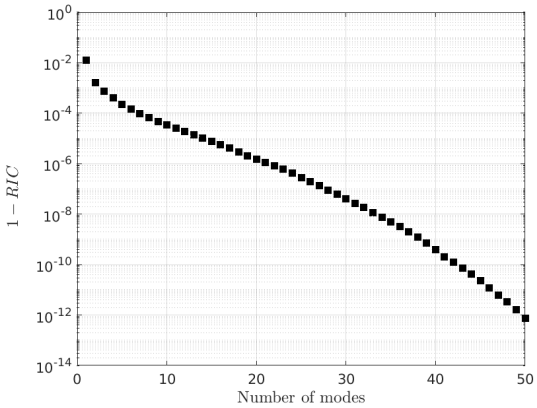

Figure 3.
SVD: plot of $k \mapsto 1 - RIC(k)$, where $RIC(k)$ represents the relative information content at truncation rank k.

thanks to the added nonlinear terms. A surprising result is that the affine ROM-A model returns rather accurate results whereas the fluid-structure interaction problem is intrinsically nonlinear.

In order to study the stability of the model ROM-A, in Figure (5) we plot the complex eigenvalues of the square matrix A_μ. We observe that all the eigenvalues have a modulus less or equal to one. One of the eigenvalues is 1 exactly (up to

DP roundoff errors), meaning that there is a physical invariant in the system. It is known that the capsule volume is kept constant during time, because of the incompressibility of the fluid. For ROM-A, since $\boldsymbol{a}_\mu^{n+1} = A_\mu \boldsymbol{a}_\mu^n + \boldsymbol{b}$ and $\boldsymbol{a}_\mu^0 = \boldsymbol{0}$, we have

$$\boldsymbol{a}_\mu^n = A_\mu^n \boldsymbol{a}_\mu^0 + \sum_{k=0}^{n-1} A_\mu^k \boldsymbol{b} = \sum_{k=0}^{n-1} A_\mu^k \boldsymbol{b}.$$

The matrix A_μ is observed to be diagonalizable in $\mathbb{C}$. There is an invertible matrix P_μ such that $A_\mu = P_\mu \Lambda_\mu P_\mu^{-1}$ where Λ_μ is the diagonal matrix of the eigenvalues. Since it is observed that $\rho(\Lambda_\mu) = 1$, we have

$$\|\boldsymbol{a}_\mu^n\| \leq \mathrm{Cond}(P_\mu)\, n\, \|\boldsymbol{b}\|, \quad \forall n \in \mathbb{N},$$

showing that the coefficients in the PCA space grow at most linearly in time.

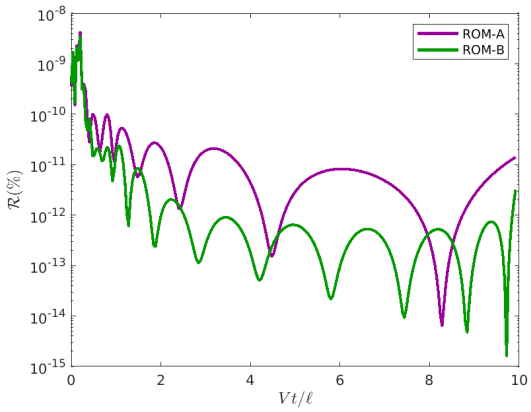

Figure 4.
Matrix identification. Log of the normalized residual $n \mapsto \log_{10}\left(\|\boldsymbol{a}_\mu^{n+1} - A_\mu \boldsymbol{\eta}(\boldsymbol{a}_\mu^n)\|^2 / \|\boldsymbol{a}_\mu^{n+1}\|^2\right)$ *for both ROM-A and ROM-B.*

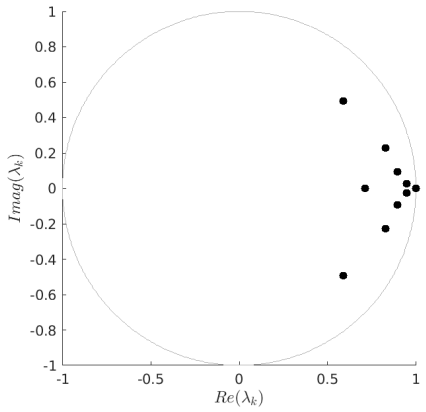

Figure 5.
Matrix identification. Eigenvalues of the computed matrix A_μ *plotted in the complex plane for the ROM-A model. One of the eigenvalue is* 1 *up to round-off error.*

In Figure 6, we compare the computed capsule shapes and positions in the channel for the computed FOM capsules obtained at different times: $t = 0, 0.4,$ 2.8, 5.2 and 7.6. What can be observed is that the ROM-B model returns very satisfactory results where the shape solutions fully overlap the FOM ones 'at the eye norm'. More finely, in Figure 7, the time evolution of the errors in the capsule 3D shape of the ROM solutions as compared to the FOM solutions is plotted. The difference between the shapes are quantified by $\varepsilon_{\text{Shape}}(t)$, the ratio between the modified Hausdorff distance (MHD) computed between the FOM shape $\mathcal{S}_{\text{FOM}}(t)$ and the ROM shape $\mathcal{S}_{\text{ROM}}(t)$ and the channel characteristic length ℓ:

$$\varepsilon_{\text{Shape}}(t) = \frac{\text{MHD}(\mathcal{S}_{\text{FOM}}(t), \mathcal{S}_{\text{ROM}}(t))}{\ell}.$$

The modified Hausdorff distance measures the maximum value of the mean distance between the two shapes to compare [16]. The ROM-A and ROM-B models return very accurate solutions with maximum 0.1% error. It is also observed that the ROM-B models is slightly more accurate than the affine approximation.

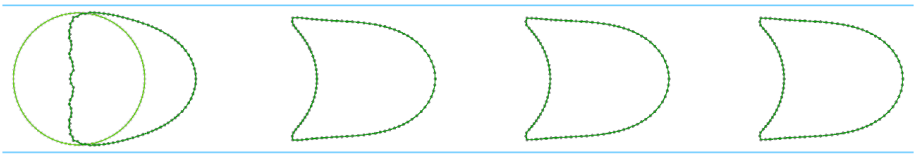

Figure 6.
Sequence of cross-section capsule shapes and positions in the microchannel from the initial spherical shape shown in light green at the beginning of the channel: comparison of the FOM solutions (grey dots) and of the solutions computed from the dynamical k-DMD reduced-order model (dark green solid line) at the same instants as in Figure 1: $t = 0, 0.4, 2.8, 5.2, 7.6$.

6.2 HOSVD on the whole data tensor, error measurements on the whole database

Now the consider the whole database made of 55 samples in the parameter domain. In Figure 8 we plot the location of the 55 chosen samples in the plane $(Ca, a/\ell)$. The design zone of interest corresponds to capsule shapes that can reach a steady state after a certain time.

A SVD is first performed on the μ-flattening of the data tensor $\mathcal{T}_x$. In Figure 9, we plot the four first parameter (normalized) eigenmodes in the parameter domain. These modes give an idea on the dependency of the capsule shapes with respect to the parameters. To complete the analysis, on Figure 10 we show the spectrum of the singular values for the μ-flattening matrix. One can observe a rather fast decay of the singular values especially for the ten first modes.

Next, we perform the SVD of the time t-flattening matrix of $\mathcal{T}_x$. The SVD provides us temporal eigenmodes. In Figure 11, the four first temporal eigenmodes ω_m, $m = 1, ..., 4$ are plotted. The first one appears to be the linear function while the others add details especially in the transient zone of the capsule dynamics. The spectrum of the singular values again shows a fast decay especially for the six first modes.

To conclude this section, we have tested the accuracy of both ROM-A and ROM-B on the whole database. For each sample, we have derived a ROM model,

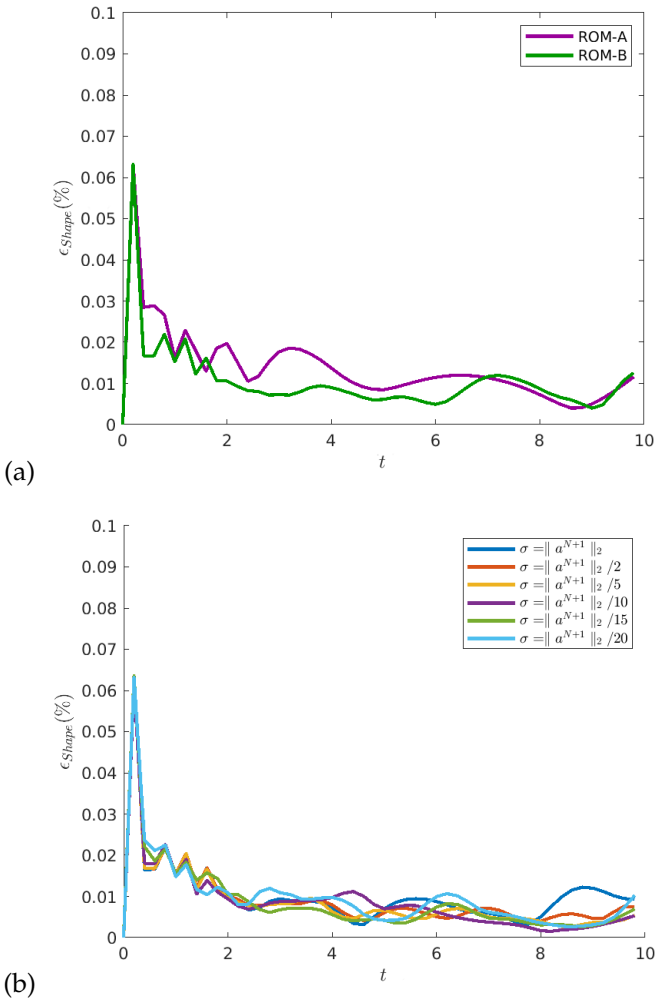

Figure 7.
(a) Comparison of the time evolution of the shape error between the affine DMD model (ROM-A with $K = 10$) and the kernel-based one (ROM-B with $K = 10$, $M = 5$). One can observe a maximum error less than 0.1% in both cases. (b) Sensitivity analysis of the parameter σ ($\| \boldsymbol{a}^{N+1} \|_2 = 667.14$.)

i.e. a low-order dynamical system formulated in the PCA space. Then we have compared the ROM solution to the FOM solution by calculating $\varepsilon_{\text{Shape}}$ between the two capsule shapes. In Figure 12, the heat maps of $\varepsilon_{\text{Shape}}$ are plotted for ROM-A and ROM-B. One can observe a uniform accuracy over the whole parameter domain, with errors less than 0.1%, thus showing the efficiency of the approach. Reported computational speedups are between 5,000 and 10,000 using ROM models. A computer with two Intel Xeon GOLD 6130 CPU (2.1 Ghz) has been used for the numerical tests.

7. Discussion

Having shown how to apply the affine DMD and *k*-DMD models and the very high precision in prediction that they offer, we now would like to provide

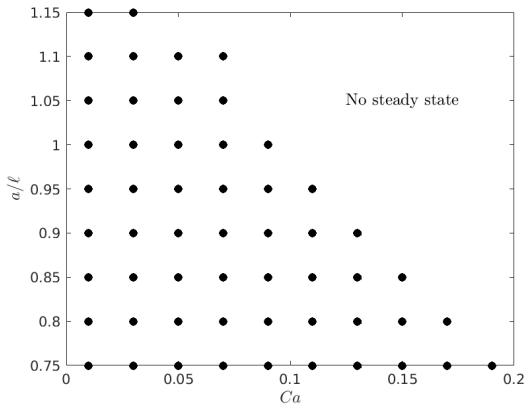

Figure 8.
Samples of the design of experiment in the parameter space $(Ca, a/\ell)$. The zone of interest corresponds to capsule shapes that reach a steady state after a certain time.

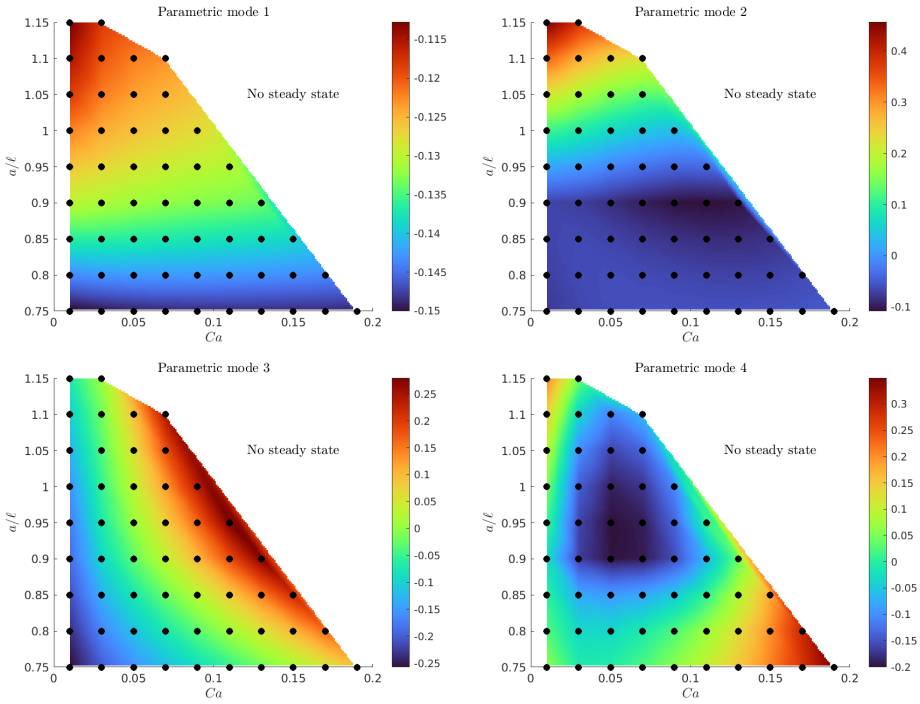

Figure 9.
HOSVD: the first four parameter eigenmodes in the parameter domain, computed from the μ-flattening of the data cube.

some further comments and remarks on model order reduction and DMD-type approaches.

7.1 Kolmogorov n-width

The method is efficient if the spectrum of singular values decays rapidly, leading to a small truncation rank K. If the spectrum decays slowly, there are two possible reasons for that: either the entropy (variety) of information in the data

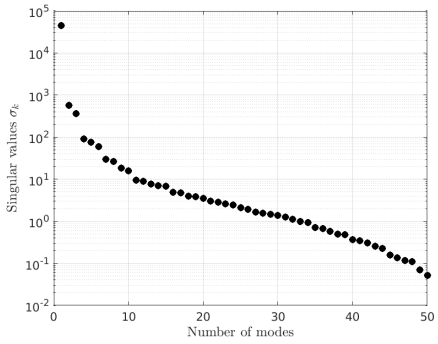

Figure 10.
Spectrum of the singular values of the μ-flattening matrix.

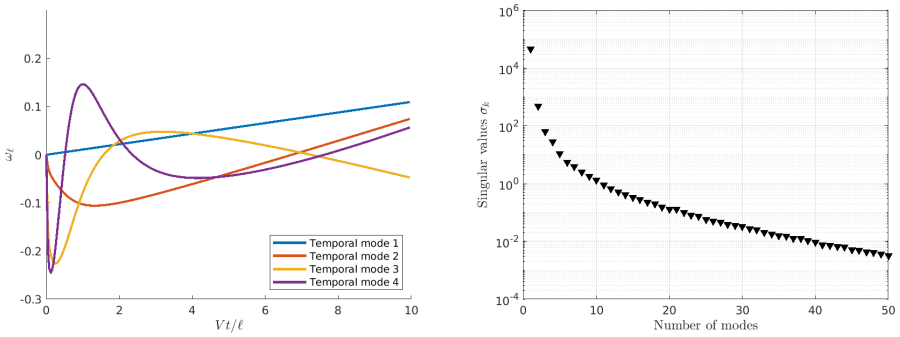

Figure 11.
HOSVD. Left: first four temporal eigenmodes computed from the SVD decomposition of the t-flattening data tensor. Right: spectrum of the singular values.

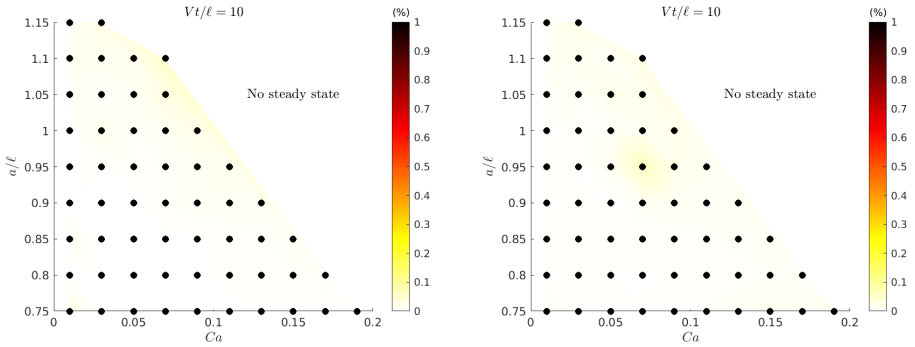

Figure 12.
Heat maps of the Modified Hausdorff distance between the FOM solutions and the ROM ones at dimensionless time $Vt/\ell = 10$. Left: ROM-A, right:ROM-B. Errors are less than 0.1%.

is high, or the solutions do not live in a linear space but rather on a nonlinear manifold. To fix the problem, one can proceed by performing a preliminary clustering of the data, scanning the parameter dimension. One can either use standard clustering techniques such as *K*-means, or a multidimensional scaling (MDS) approach. Then for each cluster, one can consider again a HOSVD and reduced-order approach suitable for each element of the cluster.

7.2 Selection of the kernel functions and kernel interpolation points

As already mentioned, the choice of the kernel function depends on the applications, on the behavior of solutions and/or on the underlying Physics. Without any a priori information, one can use universal approximation kernels like the Gaussian one. The accuracy of the results will also strongly depends on the choice of the kernel interpolation points $\boldsymbol{a}_{(j)}$. The sampling $\{\boldsymbol{a}_{(j)}\}_{j=1}^{m}$ has to correctly fill in the admissible space, or at least the state-space trajectory of interest. There are different possible strategies. A first candidate is the use of a clustering approach applied to the state-space data. The points $\boldsymbol{a}_{(j)}$ then are the centroids of each cluster. But one can consider more sophisticated approaches like a greedy iterative approach that controls the interpolation error on the data. At each iterate an interpolation point $\boldsymbol{a}_{(j)}$ is added at the location of worst interpolation error, considering all the sample solutions.

7.3 Interpretation in terms of a recurrent neural network

Let us remark that the approach can be reinterpreted as a (supervized) two-layer recurrent artificial neural network (RNN) [17]. The first layer consists in generating the features $k_i(\boldsymbol{a}_\mu)$. The second layer is a linear matrix-vector multiplication using the matrix A_μ.

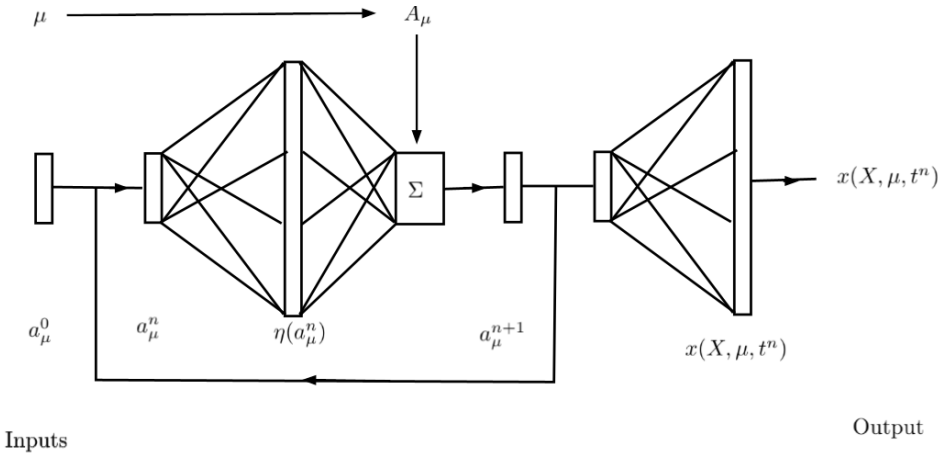

Figure 13.
Interpretation of the method as a recurrent neural network (RNN) in the PCA space.

8. Conclusions

In this chapter, the higher-order singular value decomposition has been proved to be a flexible and valuable tool in the data-driven reduced-order modeling of solutions of space-time-parameter problems, which are today at the heart of many industrial applications. The methodology has been tested on a problem of fluid-structure interaction of deformable microcapsules flowing into a microchannel. Stokes equations have been used in the fluid whereas a nonlinear hypereleastic law has been used for the membrane. Different shape solutions computed by the full-order model have been stored into a third-order tensor. First, HOSVD allows us to compute spatial, temporal and parameter

principal components and at the same time to compress the data. We get a low-order representation of the solutions with a shared spatial reduced basis. Spatial principal components are observed to provide suitable details in the shape solutions. The modes are arranged in decreasing order of importance according to the relative information content criterion. Next, additional ingredients such as kernel approximation and kernel-based dynamic mode decomposition are used to determine a reduced-order dynamical system for any parameter vector in the admissible parameter domain. The resulting low-dynamical system can be seen as an encoded recurrent neural network set into the PCA space. The approach allows us to explore the different shape solutions and visualize their evolution in the channel in real time.

Acknowledgments

The authors would like to thank Prof. Pierre Villon (UTC) for fruitful discussions on this topic. The project received funding from the European Research Council (ERC) under the European Union's Horizon 2020 Research and Innovation Programme (Grant agreement No. ERC-2017-COG - MultiphysMicroCaps).

Conflict of interest

The authors declare no conflict of interest.

Nomenclature

SVD	Singular value decomposition
HOSVD	Higher-order singular value decomposition
PCA	Principal component analysis
MOR	Model order reduction
FOM	Full order model
ROM	Reduced order model
FSI	Fluid-structure interaction
DoCE	Design of computer experiment
POD	Proper orthogonal decomposition
RIC	Relative information content
DMD	Dynamic mode decomposition
k-DMD	Kernel-based dynamic mode decomposition
MHD	Modified Hausdorff distance
RNN	Recurrent neural network

Author details

Florian De Vuyst[1*†], Anne-Virginie Salsac[2†] and Claire Dupont[2†]

1 Compiègne Applied Mathematics Laboratory, University of Technology of Compiègne – CNRS, 60203 Compiègne, France

2 Biomechanics and Bioengineering Laboratory (UMR CNRS 7338), University of Technology of Compiègne – CNRS, 60203 Compiègne, France

*Address all correspondence to: fdevuyst@utc.fr

† These authors contributed equally

References

[1] Hu X.-Q., Salsac A.-V., Barthès-Biesel D. Flow of a spherical capsule in a pore with circular or square cross-section. Journal of Fluid Mechanics. 2012, 705, 176-194. DOI:10.1017/jfm.2011.462

[2] 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. International Journal for Numerical Methods in Engineering. 2010, 83, 829–850. DOI : 10.1002/nme.2859

[3] Audouze C., De Vuyst F., Nair P.B. Reduced-order modeling of parameterized PDEs using time–space-parameter principal component analysis. International Journal for Numerical Methods in Engineering, 2009, 80, 1025-–1057.

[4] Kutz J. N., Brunton S. L., Brunton B. W. and Proctor J. L. Dynamic Mode Decomposition: Data-Driven Modeling of Complex Systems, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA. 2016, 234 p. DOI: 10.1137/1.9781611974508

[5] Tucker L. R. Some mathematical notes on three-mode factor analysis. Psychometrika. 1966, 31 (3), 279–311. DOI:10.1007/bf02289464

[6] De Lathauwer L., De Moor B., Vandewalle J. A Multilinear Singular Value Decomposition. SIAM Journal on Matrix Analysis and Applications. 2000, 21 (4), 1253–1278. DOI:10.1137/s0895479896305696

[7] Hackbusch W. Tensor Spaces and Numerical Tensor Calculus. Springer Series in Computational Mathematics. 2012, 42. DOI:10.1007/978-3-642-28027-6

[8] Grasedyck L. Hierarchical Singular Value Decomposition of Tensors. SIAM Journal on Matrix Analysis and Applications. 2010, 31 (4), 2029–2054. DOI:10.1137/090764189

[9] Vannieuwenhoven N., Vandebril R., Meerbergen K. A New Truncation Strategy for the Higher-Order Singular Value Decomposition. SIAM Journal on Scientific Computing. 2012, 34 (2), A1027–A1052. DOI:10.1137/110836067

[10] Goldfarb D., Zhiwei Q. Robust low-rank tensor recovery: Models and algorithms. SIAM Journal on Matrix Analysis and Applications. 2014, 35 (1), 225–253. DOI:10.1137/130905010

[11] Eldén L., Savas, B. A Newton–Grassmann Method for Computing the Best Multilinear Rank-(r_1, r_2, r_3) approximation of a Tensor. SIAM Journal on Matrix Analysis and Applications. 2009, 31 (2), 248–271. DOI:10.1137/070688316

[12] Brunton S. L., Budišić M. and Kaiser E. and Kutz J. N. Modern Koopman Theory for Dynamical Systems, ArXiv. 2021,abs/2102.12086

[13] Benner P., Himpe C., Mitchell T. On reduced input-output dynamic mode decomposition, Advances in Computational Mathematics. 2018, 44,1751–1768.

[14] Dupont C., De Vuyst F., Salsac. A.-V. Data-driven kinematics-consistent model order reduction of fluid-structure interaction problems: application to deformable microcapsules in a Stokes flow, Journal of Fluid Mechanics. Under revision.

[15] Fasshauer G., McCourt M. Kernel-based Approximation Methods using MATLAB. Interdisciplinary Mathematical Sciences (Volume 19). 2015, 536 p.

https://doi.org/10.1142/9335

[16] Dubuisson M., Jain A. K. A modified Hausdorff distance for object matching. Proceedings of 12th International Conference on Pattern Recognition. 1994, 1, 566-568.

[17] Tsoi A. C. and Back A. Discrete time recurrent neural network architectures: A unifying review. Neurocomputing. 1997, 15, (3–4),183-223. https://doi.org/10.1016/S0925-2312(97)00161-6

[18] Boubehziz T., Quesada-Granja C., Dupont C., Villon P., De Vuyst F., Salsac A.-V. A Data-Driven Space-Time-Parameter Reduced-Order Model with Manifold Learning for Coupled Problems: Application to Deformable Capsules Flowing in Microchannels. Entropy, 2021, 23, 1193. https://doi.org/10.3390/e23091193