Learning slosh dynamics by means of data

B. Moya[1] · D. Gonzalez[1] · I. Alfaro[1] · F. Chinesta[2] · E. Cueto[1]

Abstract

In this work we study several learning strategies for fluid sloshing problems based on data. In essence, a reduced-order model of the dynamics of the free surface motion of the fluid is developed under rigorous thermodynamics settings. This model is extracted from data by exploring several strategies. First, a linear one, based on the employ of Proper Orthogonal Decomposition techniques is analyzed. Second, a strategy based on the employ of Locally Linear Embedding is studied. Finally, Topological Data Analysis is employed to the same end. All the three distinct possibilities rely on a numerical integration scheme to advance the dynamics in time. This thermodynamically consistent integrator is developed on the basis of the General Equation for Non-Equilibrium Reversible–Irreversible Coupling, GENERIC [M. Grmela and H.C Oettinger (1997). Phys. Rev. E. 56 (6): 6620–6632], framework so as to guarantee the satisfaction of first principles (particularly, the laws of thermodynamics). We show how the resulting method employs a few degrees of freedom, while it allows for a realistic reconstruction of the fluid dynamics of sloshing processes under severe real-time constraints. The proposed method is shown to run faster than real time in a standard laptop.

Keywords Data-driven fluid simulation · Model order reduction · GENERIC formalism · Real-time simulation

1 Introduction

The interest for fast and realistic fluid simulation—noteworthy, free-surface phenomena [13]—is due to the wide range of applications that could derive from their results. Some examples could be the improvement of graphics for computer games, surgery simulation or virtual prototyping, to name but a few. Computer graphics is one of the areas seeking to reproduce complex behaviors, traditionally studied through the Navier Stokes equations, in order to provide convincing representations of different case studies that range from flow divergence to splatter [2,27].

Nonetheless, real-time simulation of free surface fluid flows is still an open research field. The main issue which we address here lies in the commitment between accuracy and computation time. Simulations of this kind are computationally demanding, and thus difficult to perform in such conditions to produce credible results. In addition, a physically consistent model from which we could extract valuable information would be highly desirable. The simulation-based control of robots manipulating liquids, for instance, is of utmost societal importance and still presents important difficulties [38–40]. Therefore, there is room for new techniques to be developed able to fulfill these prerequisites.

Recently, there is an increasing interest in data-driven simulation so as to find appealing alternatives to this problem, common to both solid and fluid mechanics [5,10,11,14,16,17,21,24,31]. This approach is particularly interesting in the field of robotics, where data coming from computer vision is to be analyzed so as to provide the necessary control feedback through simulations in the loop. Machine and deep learning methodologies are some of the means employed to unveil data structures in order to develop predictive algorithms. They provide not only a realistic representation of the fluid, but also the properties of the physical state as outcome.

✉ E. Cueto
ecueto@unizar.es

B. Moya
beam@unizar.es

D. Gonzalez
gonzal@unizar.es

I. Alfaro
iciar@unizar.es

F. Chinesta
Francisco.Chinesta@ensam.eu

[1] Aragon Institute of Engineering Research, Universidad de Zaragoza, Zaragoza, Spain

[2] ESi Chair and PIMM Lab. Ensam ParisTech, Paris, France

Ladicky [23] presented a GPU implementation consisting of a forest regressor [4], trained with a large set of videos to extract pairs of snapshots, able to predict the state of each particle in the following time step, characterized by the so-called feature vector of state variables. This algorithm allows to not be limited by the size of the time step, considered the bottleneck of the simulation. Closely related, Byungsoo [20] explored reduced representations of smoke and different types of liquids by means of deep learning. They constructed non-linear functions through the use of convolutional networks from the velocity fields so that the reduced subspace was better adapted to the movement compared with the results obtained by lineal methods.

In a related setting, there has been a growing interest on the development of techniques based on the Koopman operator and the related Dynamic Mode Decomposition (DMD) method [18,22,32]. DMD computes a set of space-time modes—in opposition to classical proper orthogonal decomposition (POD) techniques, that compute only spatial modes—based on snapshots of the velocity field that are in fact the eigenmodes and eigenvalues of a linear approximation to the nonlinear governing operator of the fluid dynamics [1,33].

Following a similar rationale, the work of Kevrekidis and coworkers on *equation free modeling,kevrekidis2010 equation* also starts from data and constructs an approximation of the *slow manifold* on which the system lives. All of these methods asume that the dynamics of the problem is governed by an equation whose precise form is hard to find or, simply, we are not interested in.

The present work resembles somehow the strategy developed by Millan and Arroyo [26]. In that case, emphasis was put in the non-linear dimensionality reduction aspects of the technique. As will be done here, Millan and Arroyo developed an integration scheme in generalized (reduced) coordinates. A fundamental difference of the present work, as will become clear readily, is that the integration scheme will be learnt from data and—contrarily to the work of Millan and Arroyo—is valid for Hamiltonian as well as for non-conservative or out of equilibrium mechanics.

In this work we focus on sloshing problems, whose interest—particularly for the control of robots manipulating liquids—has already been mentioned. The method thus developed is based on two important assumptions. First, we work with data [3], but we are interested in maintaining as much as possible of the well-known mathematical description of the physical laws governing the problem. Indeed, we are particularly interested in satisfying the basic principles of thermodynamics—conservation of energy and positive production of entropy—so as to render the method consistent and also to obtain a sound numerical method with the right stability properties, despite the presence of noise in the data. Second, as in the equation-free modeling setting [19], we assume the existence of a slow manifold in which the fluid evolves. The method will be thus able to find the geometrical structure of this slow manifold based solely on data, while providing with a numerical method for the approximate integration of the dynamics on this manifold. To this end, given the inherent dissipative structure of most fluids, we resort to a general enough description, which in this case will be provided by the GENERIC framework [12,30]. GENERIC constitutes a generalization of the Hamiltonian physics to dissipative (non-equilibrium) systems. It is, therefore, the most general structure to be fulfilled by our method so as to satisfy the basic principles of thermodynamics. Noteworthy, finite difference discretization in time of GENERIC also provides with the right structure of a numerical integration scheme so as to satisfy these principles for long-time integration times [34].

The structure of the paper is as follows. Section 2 consists of a discussion about the model order reduction of sloshing phenomena. For this purpose, three model order reduction techniques have been tested: POD, locally linear embedding (LLE) [37] and topological data analysis (TDA) [42]. This last approach is found to be the most robust, so that the final method relies upon it. GENERIC is presented in Sect. 3 as an integrator for sloshing dynamics in a thermodynamically consistent framework. The outcome aims to become a powerful CPU system able to work at real time rates. Section 4 will consist of a review of the numerical results obtained by the implementation and test of the integration algorithm, to end up with a discussion in Sect. 5 over its efficiency and future improvements.

2 Finding the slow manifold

In a general setting, we are interested in systems whose evolution in the time interval $\mathcal{I} = (0, T]$ will be governed by a set of variables $z_t = z(t) : \mathcal{I} \to \mathcal{S}$, $z \in \mathcal{C}^1(0, T]$, i.e.,

$$\dot{z}(\boldsymbol{x}, t) = \frac{dz}{dt} = f(z(\boldsymbol{x}, t)),$$

where $\boldsymbol{x}$ represents the physical coordinates of the considered point. Given the expected complexity of f, and also with an eye towards the future use of the method here developed in the control of robots, we are interested in reverse engineering this governing equation by means of data. We are therefore not interested in obtaining a closed-form expression of f, but to be able to integrate it in time with reasonable accuracy and, above all, under severe real-time constraints.

As mentioned in the introduction, many previous works hypothesize the existence of a slow manifold to which the dynamics of the system tends [19,22]. The objective of this work, therefore, will be first to unveil the geometrical struc-

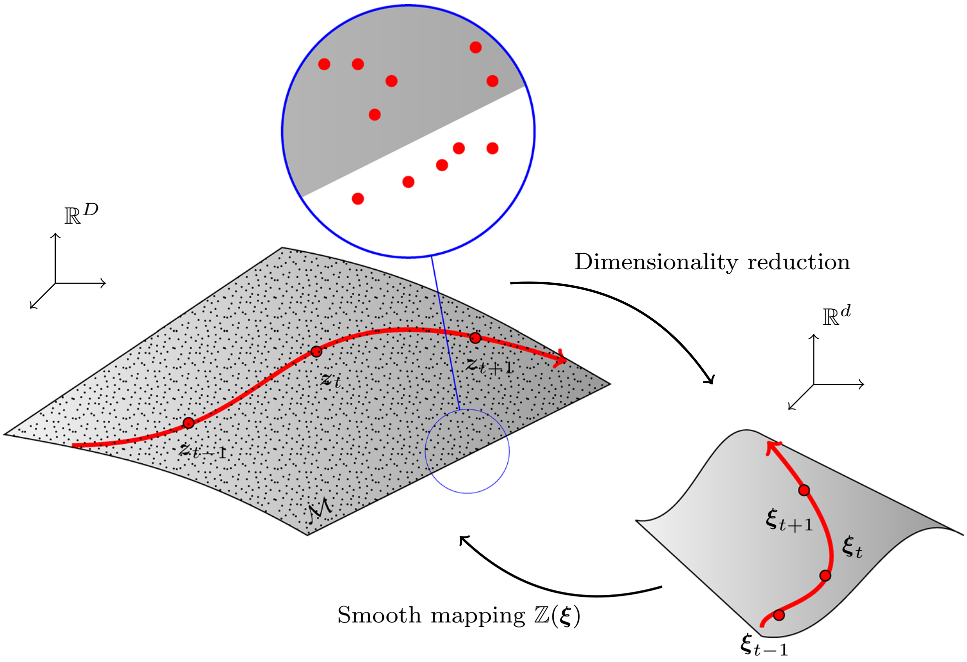

Fig. 1 Hypothesis abut the existence of a slow manifold $\mathcal{M}$ on which the fluid lives. The small dots represent the experimental data (snapshots) in a high-dimensional space $\mathbb{R}^D$. These (possibly noisy, see the zoomed detail) data are assumed to describe a manifold whose geometry is sought. An arbitrary trajectory of the system in the phase space is represented in red. Given the high-dimensionality of the sought manifold, a dimensionality reduction will also be applied so as to project the data to an embedding space in $\mathbb{R}^d$, with $d \ll D$. Then, the system dynamics will be integrated in this embedding space and mapped back to the physical space in $\mathbb{R}^D$

ture of this manifold, see Fig. 1, and then, to be able to integrate the equations of motion on it.

Arroyo and coworkers also accepted the existence of a slow configuration manifold for finite strain hyperelastodynamics and developed a method for the integration of the equations of motion, described by a minimal set of degrees of freedom [26]. It is well-known that in this case (a purely Hamiltonian case) positions and momenta describe the cotangent bundle of the configuration manifold, $T^*\mathcal{M}$, and therefore the existence of this manifold is guaranteed [7].

In the case of (viscous) fluids, we are not in a Hamiltonian framework, given the dissipative character of viscous phenomena. Therefore, the problem will involve, on one hand, the need for a rigorous choice of the set of variables z describing the configuration manifold. On the other, given their infinite-dimensional character—note the dependence of $z(\boldsymbol{x}, t)$ on $\boldsymbol{x}$—it will be necessary to discretize it by employing a minimal number of degrees of freedom. In the work of Arroyo this was done by employing non-linear dimensionality reduction techniques to finite element results for a collection of problems [26].

In what follows we discuss the approach followed herein. We begin by the description of the pseudo-experimental campaign. We then introduce different possible approaches to characterize the fields of variables z.

2.1 Obtention of pseudo-experimental results

Our proof of concept is a viscous fluid in a container of dimeter $b = 10$ cm and height $h = 7$ cm, which has been subjected to sloshing forces, see Fig. 2. In order to conduct the reduction of the dynamics, the Navier–Stokes equations governing the fluid motion have been first discretized by applying a Smooth Particle Hydrodynamics approach [9]. Particle discretization has been largely applied to analysis since particle-based methods offer good approximations to deal with complex fluids due to its efficient adaptability to complicated geometries.

It is well-known that, for a Newtonian fluid, the adequate set of state variables that characterize a fluid particle is composed by its position, velocity and internal energy [8], so that

$$\mathcal{S} = \{z = (\boldsymbol{r}_j, \boldsymbol{v}_j, E_j, j = 1, 2, \ldots, M) \in (\mathbb{R}^3 \times \mathbb{R}^3 \times \mathbb{R})^M\}, \tag{1}$$

where M represents the number of particles in the SPH discretization. The fluid of our case of study is discretized by 2898 particles.

These pseudo-experimental results constitute a huge amount of data. Reconstructing the dynamics of the flow in such a high-dimensional space would need for a tremendous amount of memory and CPU time. As sketched in Fig. 1, our aim is to reduce the dimensionality of the space in which the dynamics are to be integrated. Thus, given a matrix composed by the set of n snapshots grouped by columns,

$$\begin{bmatrix} | & | & & | \\ z_1 & z_2 & \cdots & z_n \\ | & | & & | \end{bmatrix} = \boldsymbol{Z} \in \mathbb{R}^{D \times n},$$

that describe the time history of the sloshing movement in $D = 7 \times M$ dimensions, representing the state of each par-

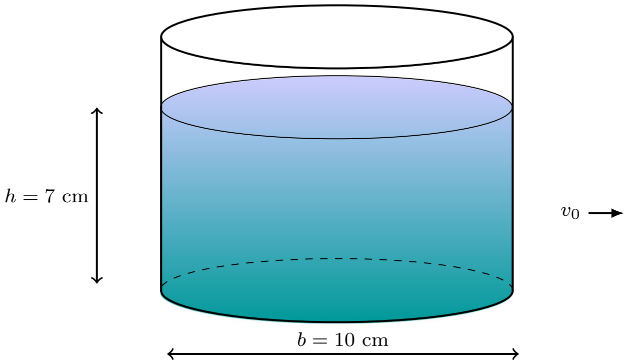

Fig. 2 Geometry of the liquid container

ticle at each time step $t \in [t_1, \ldots, t_n]$, we aim to embed the slow manifold of the system in dimension d so that $d \ll D$. Since in this particular case we have seven relevant variables to describe the state of each particle, the dimension D of each snapshot vector of the SPH approximation to the problem is therefore $2898 \times 7 = 20{,}286$, justifying the need for a re-parametrization so as to embed the dynamics in a lower dimensional manifold. The selection of this new parametrization will be further discussed in the next sections, where different methods will be discussed.

The experimental campaign is composed by four simulations of sloshing dynamics at incrementally higher initial velocities $v_0 = \{0.05, 0.15, 0.1, 0.2\}$ m/s. The set of snapshots obtained by a commercial simulation software has been calculated for a time increment of $\Delta t = 0.00016$ s. In total, we have obtained 10625 snapshots for each simulation. Conversely, we do not need the full set of snapshots. The set will be pruned by selecting equally spaced snapshots to keep the minimum required to capture the dynamics. With regard to the reduction method imposed, more or less snapshots will be required.

It must be noticed that the time integration of the dynamics will be preformed in the slow manifold, after a judicious parametrization by embedding it in a low-dimensional space. Therefore, a mapping is required to connect the slow manifold and its embedding in the low-dimensional space. To this end, several strategies have been analyzed that are discussed next.

2.2 A POD approach

Proper orthogonal decomposition [25], has been largely applied to dynamical problems to find an optimal orthogonal projection of the system in which a good approximation of the dynamics can be defined. The motivation of its application over our sloshing problem yields in the fact that we consider that our problem is likely to present coherent structures, which means that we can find statistical relationships between variables.

Method of snapshots. In our case, since we have performed a large discretization of the fluid, the dimensionality of the problem is much greater than the available collection of snapshots. As a result, the data matrix over which we perform POD is narrow but large. If we did an ordinary analysis, the computation of the correlation matrix $\boldsymbol{Q} = \boldsymbol{Z}\boldsymbol{Z}^\top \in \mathbb{R}^{D \times D}$ would be computationally demanding. In such cases, the use of the method of snapshots is more convenient. The correlation matrix would be calculated as $\boldsymbol{Q} = \boldsymbol{Z}^\top \boldsymbol{Z} \in \mathbb{R}^{n \times n}$ instead, to carry out the eigenvalue decomposition

$$\boldsymbol{Q}\boldsymbol{v}_j = \lambda_i \boldsymbol{v}_j, \;\; j = 1, \ldots, n,$$

being λ_i the set of eigenvalues and $\boldsymbol{v}_j \in \mathbb{R}^n$ the eigenvectors of the analysis. These eigenvectors are the right singular vectors of the singular value decomposition. Therefore, they are equivalent to the corresponding POD modes, which can be simply obtained by the following transformation

$$\boldsymbol{u}_j = \frac{1}{\sqrt{\lambda}} \boldsymbol{Z}\boldsymbol{v}_j \;\; j = 1, \ldots, n.$$

Large eigenvalues represent the main hidden variables of the system, and the smallest are related to perturbations. We aim to choose the minimum dimension capable of representing the dynamics accurately. We need a reduced-basis coordinate system able to capture most of the energy of the system. We will select a number d of orthogonal modes able to represent a prescribed amount of the energy of the system. Conversely, due to the high non-linearity of the system, we observe that a number of modes higher than that value may be needed to ensure that we have an accurate integration scheme. This set of d vectors conforms the projection matrix $\boldsymbol{D}$, that works bidirectionally to offer a lineal mapping between the physical space $\boldsymbol{Z}$ and the reduced basis $\boldsymbol{\Xi}$,

$$\boldsymbol{Z} = \boldsymbol{D}\boldsymbol{\Xi}. \tag{2}$$

The problem with a POD-based approach is that it provides with the best *linear* dimensionality reduction for the set of snapshots. In this case, the dynamics of the liquid will be non-linear and a big number of terms in the basis is expected, thus rendering the method less efficient.

2.3 A locally linear embedding approach

Some phenomena are poorly described by linear methods since, due to the high non-linearity of their behavior, the intrinsic relationships are also non-linear and hard to be described by linear correlations. Thus, new manifold learning methodologies have emerged to face these shortcomings. Locally Linear Embedding, known as LLE [37], considers that each point of the cloud will be interpolated by its K nearest neighbors with regard to their pairwise euclidean distance. Local linearity will be assumed to exist in these patches of the manifold. We take into consideration the premise that the number of neighbors should be greater than the estimated intrinsic dimensionality d of the slow manifold at the time to select these parameters. In contrast, due to the complex description of our case of study, we may need a higher dimensionality d to capture the main features of the whole dynamics.

In order to obtain an embedding of the system onto a lower dimensional space, we proceed in three steps. First, we find the K-nearest neighbors of each point. Then, we compute the reconstruction of each point from its neighbors. For this purpose, we calculate the weights that enable the interpolation through the minimization of the approximation error, measured by the cost function represented by

$$\varepsilon(\boldsymbol{W}) = \sum_{i=1}^{n} \left| \boldsymbol{z_i} - \sum_{j=1}^{K} W_{ij} \boldsymbol{z_j} \right|^2 , \tag{3}$$

restricted by two constraints. Firstly, each data point is reconstructed only by its neighbors, enforcing $W_{ij} = 0$ if $\boldsymbol{z}_j$ is not a neighbor. Secondly, weights must sum to one: $\sum_j W_{ij} = 1$. Finally, an eigenvalue problem will provide the non-linear projection of the system analyzed.

It is important to note that in LLE, the number of neighbors K is user-defined and constant for every data point. This is a reasonable assumption if the sampling is dense and a small number of neighbors is chosen. However, it does not take into account the true topology of the data or how intricate the manifold is at different regions. Indeed, as an assumption, the linearity condition imposed at each patch is also considered to hold in the embedded manifold. Therefore, the weights remain constant at any subspace of dimension $d \ll D$. In other words, they are invariant to transformations such as translation, rotation and scaling, enabling a linear mapping between the low dimensional space and the high dimensional system. Then, the only unknown are the new coordinates $\boldsymbol{\Xi} = \{\boldsymbol{\xi}_i\}$, $i = 1, \ldots, M$ of the points in the embedded space, which are calculated by the cost function related to the embedded space,

$$\Phi(\boldsymbol{\Xi}) = \sum_{i=1}^{n} \left| \boldsymbol{\xi_i} - \sum_{j=1}^{K} W_{ij} \boldsymbol{\xi_j} \right|^2 . \tag{4}$$

This results in a $D \times D$ eigenvalue problem, whose d-bottom non-zero eigenvectors represent a set of optimal coordinates in which the manifold is embedded. Given the local linearity constraint, if we introduced a new point in the slow manifold, it could be locally linear interpolated by a set of neighbors in the subspace and, with the calculated coefficients and the set of neighbors, reconstruct it in the complete space.

2.4 A TDA approach

Data science has recently be attracted by the sought of topological underlying structures in data to analyze its complexity. One of the milestones achieved by these techniques is its use for manifold learning purposes. Data has a shape, which can be interpreted as the slow manifold where it is embedded. One of main strengths of Topological Data Analysis (TDA) is that it makes data less sensitive to noise or outliers by discerning the intrinsic features of the set of data. Due to the reduction capacity of this technique, TDA can be used to visualize high dimensional data or as a preprocessing method for supervised manifold learning methodologies in order to find the smooth manifold of the object of study.

TDA makes an intensive use of persistent homology [28, 42], which consists in the sought of topological features of data. Information can be extracted from a point cloud just by analyzing its clustering history. The study is performed through the selection of a connectivity parameter, which will determine the shape of data. The internal elements we refer to are known as *simplices*. For a given connectivity parameter R, two points are pairwise connected if they are far apart at most R. As a consequence, they will conform a 1-simplex, or edge. To define a simplex of higher dimensionality, the M-points that would conform it must be closer than R to every other point. The join of various *simplices* results in a *simplicial complex*.

The final goal of the analysis is to construct a simplicial complex able to characterize and represent accurately the structure data. The optimal parameter R is selected from the persistence diagrams. Persistence diagrams are a representation of the time history of the elements. They show barcodes, that show when the simplicial complex appears and disappears along time. Short barcodes show structures that are interpreted as noise, while long bars show the predominant structures of data. Therefore, R will ensure that the result avoids noise and represents the natural underlying shape of the point cloud. For a graphical explanation of TDA, see Fig. 3.

The reader will notice how, by employing big (and high-dimensional) data, the time taken by TDA methods to compute neighborhoods grows exponentially. To overcome this difficulty, we have employed the method of landmarks [6]. Basically, this method samples the data set so as to obtain a topology entirely equivalent but at a portion of the computational cost.

Once we have selected the optimal connectivity parameter R that reflects the true shape of the data set, the obtained simplicial complex storages the relationships among data

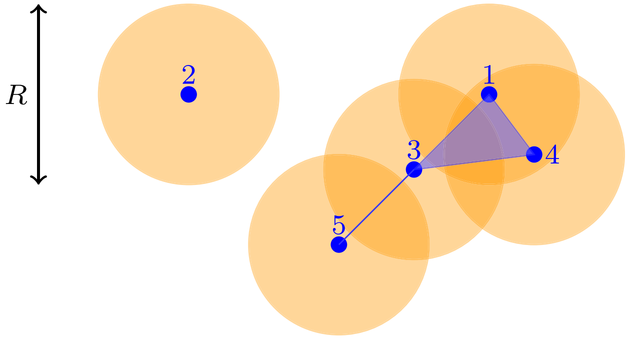

Fig. 3 A sketch of the concept of persistence homology. By making R grow from 0, simplices (edges, triangles, tetrahedra) appear (respectively, when two, three or four circles/spheres intersect) or disappear (when a hole collapses). Persistence diagrams collect the values of R for which these simplices appear or disappear. For the represented value of R, node 2 is isolated from the rest of data points, while nodes 3 and 5, for instance, are connected—their 1d persistence diagram takes a value different from zero—. On the contrary, the triangle 1-3-4 is closed for this R value, thus meaning that it must disappear from the 1d persistence diagram—a hole has collapsed—.Those simplices with higher persistence intervals represent the overall *shape* of the data set (they *persist* for wider R intervals), while brief intervals are usually associated to noise in the data

points, which represent the neighborhood of each one. In contrast with Locally Linear Embedding, the analyst does not enforce each datum to be related to a specified and fixed number of neighbors K, but to have relationships only with the *persistent* neighboring points—those data points whose neighborhood is obtained on the manifold and not the Euclidean space—. On the other hand, TDA does not provide (as LLE does) with an interpolation scheme within the just obtained manifold structure of the data. Here, we will employ Kriging [43]. Of course, it is always possible to employ the weights provided by LLE with a variable number of neighbors, but Kriging has demonstrated in our experience to provide slightly more accurate results.

From this point, we proceed like in an LLE problem, but with a variable number of neighbors for each data point, $\mathcal{N}(i)$. We calculate the embedded coordinates $\boldsymbol{\xi}_i$ of each point by minimizing the functional

$$\Phi(\boldsymbol{\Xi}) = \sum_{i=1}^{n} \left| \boldsymbol{\xi}_i - \sum_{j=1}^{\mathcal{N}(i)} W_{ij}^* \boldsymbol{\xi}_j \right|^2 ,$$

where W_{ij}^* represent the weights provided by a Kriging scheme constructed on the $\mathcal{N}(i)$ neighbors $z_i \in \mathbb{R}^D$.

It is worth noting that, as in LLE techniques, we hypothesize that the Kriging scheme is valid on the physical as well as in the embedded spaces, so that it provides with the necessary smooth mapping between the embedded coordinates and the physical ones:

$$\mathbb{Z} : \mathcal{A} \subset \mathbb{R}^d \longrightarrow \mathbb{R}^D$$
$$\boldsymbol{\xi} \longmapsto \sum_{j=0}^{\mathcal{N}(i)} W_{ij}^*(\boldsymbol{\xi}) z_j ,$$

where $\mathcal{A} \subset \mathbb{R}^d$ is the convex hull of the neighbors of point $\boldsymbol{\xi}$.

3 Integration in the slow manifold

So far, we have described how to obtain, from data, the slow manifold in which the system evolves and how to embed it in a low-dimensional space so as to speedup the process. However, a final ingredient is still missing: how to integrate the dynamics of the system on the embedded manifold. In [26] a scheme is developed by taking advantage of the Hamiltonian nature of hyperelasticity. Here, on the contrary, we develop a completely general method that, based on the data, unveils the underlying governing equations, while providing a very efficient scheme for its integration in time.

To that end we will employ the so-called General Equation for Non-Equilibrium Reversible–Irreversible Coupling, GENERIC, formalism [29,30], which can be advantageously employed to characterize any non-equilibrium thermodynamical behavior by means of data from the essential and measurable state variables of the system [10].

GENERIC constitutes a generalization of the Hamiltonian formalism of physics to non-equilibrium thermodynamic settings. Therefore, it is seen as the most general framework in which we can embed our system so as to satisfy basic principles such as conservation of energy and strictly positive dissipation of entropy. Data will be employed to unveil the different terms in its expression. As will be noticed, each constituent of the GENERIC formalism is assumed to have its own manifold structure which is to be found by the methodology introduced in the following section.

3.1 An introduction to GENERIC

The GENERIC formalism [12] is a coarse-grained thermodynamic description of non-equilibrium systems which characterizes their state from the analysis of the evolution of energy and entropy. It is a powerful tool for treating systems of practical interest since it guarantees the no violation of thermodynamic laws while representing the model with fewer degrees of freedom. The generalization offered by this formalism comes from only considering the so-called slow variables, which will persist in the stationary state, and thus characterize with stronger influence the evolution of the dynamics. This formalism obeys the first and second laws of thermodynamics due to the fulfillment of Noether's theorem, i.e., it preserves the symmetries of the system. The expression of this formalism is the following:

$$\frac{dz}{dt} = \boldsymbol{L}\frac{\partial E}{\partial z} + \boldsymbol{M}\frac{\partial S}{\partial z}, \tag{5}$$

where E represents the energy of the system and S its entropy. It can be appreciated that this expression encloses two differentiated parts to represent any thermodynamic system. $\boldsymbol{L}$ is the so-called Poisson matrix and $\boldsymbol{M}$ the friction matrix, which are related to the conservative and dissipative parts of the system respectively. In order to guarantee the conservation of energy and non-decreasing entropy conditions, the next two constraints, called degeneracy condition, must be accomplished:

$$\boldsymbol{L}\frac{\partial S}{\partial z} = \boldsymbol{0}, \qquad \boldsymbol{M}\frac{\partial E}{\partial z} = \boldsymbol{0}. \tag{6}$$

In order to fulfill these prerequisites, $\boldsymbol{L}$ must be skew-symmetric, while $\boldsymbol{M}$ must be symmetric, positive semi-definite. As a result, the conservation of energy and entropy generation are guaranteed. Given the measurements of the state of the system $\mathcal{S}$ at different discrete time steps, $\boldsymbol{Z}$, we could obtain also in discrete form the elements $\boldsymbol{L}, \boldsymbol{M}, \nabla E$ and ∇S by performing a regression process over the discretized expression of Eq. (5),

$$\frac{z_{n+1} - z_n}{\Delta t} = \mathsf{L}(z_{n+1})\mathsf{DE}(z_{n+1}) + \mathsf{M}(z_{n+1})\mathsf{DS}(z_{n+1}). \tag{7}$$

The aim of our work is to obtain the values of DE and DS, discretized energy and entropy gradients, and L and M if they were also unknown—very often they have a pre-defined structure—. As a final goal, we aim to construct the constitutive manifold of the sloshing dynamics. This concept, introduced by [15], proposes a strictly numerical approximation to work on the manifold of the latent parameters that govern the dynamics instead of relying on constitutive equations to extract results. By constructing the constitutive manifold of {L, M, DE, DS} we are developing the basis of the integration scheme for the sloshing problem. Considering the finiteness of the variables at discrete time steps, gradients operators can be approximated as:

$$\mathsf{DE} = \boldsymbol{A}z, \quad \mathsf{DS} = \boldsymbol{B}z,$$

as usual in the finite element community.

In order to determine the numerical value of each GENERIC constituent, we accomplish a regression over discrete time intervals $\mathcal{J} \subset \mathcal{I}$,

$$\boldsymbol{\mu}^* = \{\mathsf{L}, \mathsf{M}, \boldsymbol{A}, \boldsymbol{B}\} = \arg\min_{\boldsymbol{\mu}} ||z(\boldsymbol{\mu}) - z^{\mathrm{meas}}||, \tag{8}$$

subjected to the constraints

$$\mathsf{L} \cdot \boldsymbol{B}z = \boldsymbol{0},$$
$$\mathsf{M} \cdot \boldsymbol{A}z = \boldsymbol{0},$$

that ensure, respectively, conservation of energy and non-negative entropy evolution.

This methodology has already successfully constructed the constitutive manifold of dynamics in a variety of cases [10].

3.2 GENERIC form of the sloshing problem

The selection of the variables that determine the state of the particles, and hence will define the dynamics of the phenomenon, is not a trivial question and has been deeply discussed in previous works such as [8,34–36]. An appropriate selection is required to ensure the thermodynamic definition of the system. In other words, the energy $E(z)$ must be written as a function of the selected variables. If that is not possible, we lack a real GENERIC structure. In addition, two more constrains need to be added to this selection. First, variables must be accesible, i.e., observable. Secondly, they have to be independent—even if linearly-dependent variables will not affect the results, see [8]—. Following the description of a fluid through its continuum governing equations, the system is fully described by the position of the particles $\boldsymbol{r}_i$, velocity $\boldsymbol{v}_i$ and internal energy E_i of each particle i. Thus, the state of one particle is thus fully described by seven degrees of freedom,

$$z_i = \begin{pmatrix} \boldsymbol{r}_i \\ \boldsymbol{v}_i \\ E_i \end{pmatrix},$$

for each particle $i = 1, \ldots, M$ in the SPH model or, equivalently, for any experimental measurement.

From this description, we can deduce the manifold structure of L and M as functions of z. The dynamics we try to study also include a non-conservative behavior that governs the movement of the fluid.

Once the ingredients of the GENERIC expression for the problem at hand is available, Eq. (7) allows us to obtain a powerful time integrator whose numerical properties have been extensively studied in previous works, see [34–36].

4 Numerical results

In this section we will expose the results obtained from the implementation of the just introduced algorithm under the three different approaches already mentioned. Regarding the results extracted from the application of each methodology,

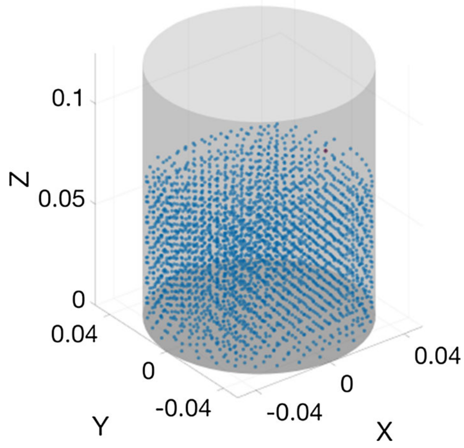

Fig. 4 Cloud of nodes of the liquid at rest. There are 2898 particles in the model. Particle number 1, for which some results are analyzed hereafter, is highlighted in red

we will analyze their efficacy in order to reach the finest degree of accuracy possible without neglecting the real-time constraint to culminate the development of the integration scheme that is, ultimately, the main milestone of this paper.

As mentioned earlier, the proof-of-concept problem that we have considered is discretized by a total number of 2898 SPH particles. Their configuration at rest is shown in Fig. 4.

In the subsequent sections, we verify the developed method by reconstructing one of the experiments (in particular, the one that corresponds to an initial velocity of 0.15 m/s). The ability of the method to reconstruct one of the experiments will provide important insight on the accuracy of the scheme.

4.1 Verification of the POD approach

Proper orthogonal decomposition is the first approach to test. In order to apply this linear strategy, we have started by selecting equally spaced snapshots at intervals of $\varDelta t = 0.02\,\mathrm{s}$. Additionally, the structure of the GENERIC matrix $\boldsymbol{L}$ is known [41],

$$\boldsymbol{L}_i = \begin{pmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ -1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & -1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & -1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & -1 & -1 & -1 & 0 \end{pmatrix},$$

so that only the discrete matrix form of the gradients $\boldsymbol{A}$ and $\boldsymbol{B}$ and the dissipative matrix M are to be identified.

From the pseudo-experimental results, it can be noticed that the eigenvalues of their POD decomposition shows a typical pattern. In it, see Fig. 5, it can readily be noticed how the first modes present an abrupt decay of several orders of magnitude. However, with 25 modes, only 62.81% of energy of the system is captured. By increasing the number of modes selected, we reach some 77.43% of energy captured with 50 modes, that is still not enough to build accurately the slow manifold of the whole set. A further increase in the number of modes renders the calculation too computationally demanding to be performed.

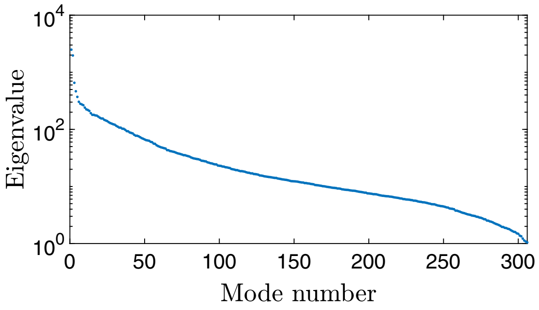

Fig. 5 Evolution of the eigenvalues of the pseudo-experimental results

When analyzing the eigenvalues of the problem we can appreciate that, although there are some modes that stand out, a great number of them are significant for representing the dynamics accurately. As a consequence, a high number of modes will be also required to capture the essential dynamics of the sloshing movement. After comparing the results for a range of modes, we conclude that 25 are enough to perform a reasonable GENERIC calculation and generate a credible representation of the fluid for each trajectory. Notwithstanding the mentioned difficulties—not surprising, given the linearity of POD method—the L_2 norm error in the reconstruction of the velocity field, see Fig. 6, is in the order of 1.5%.

It is worth mentioning that these difficulties of the POD approach could not be attributed to the GENERIC integration scheme—whose stability has been deeply studied and proved in the literature, see for instance [34]—but to the big number of POD modes necessary to obtain an accurate representation of the non-linear character of the pseudo-experimental results.

4.2 Verification of the LLE approach

Through the application of LLE, a lower number of modes is expected to capture the essential dynamics and obtain a consistente form of the GENERIC gradients. More specifically, we have applied satisfactorily the identification algorithm in an embedding manifold of dimensionality 18 for a proper reconstruction.

In this case, the L_2-norm error in the reconstruction of the velocity field significantly decreases with respect to the POD approach. This error resulted to be on the order of 0,017%.

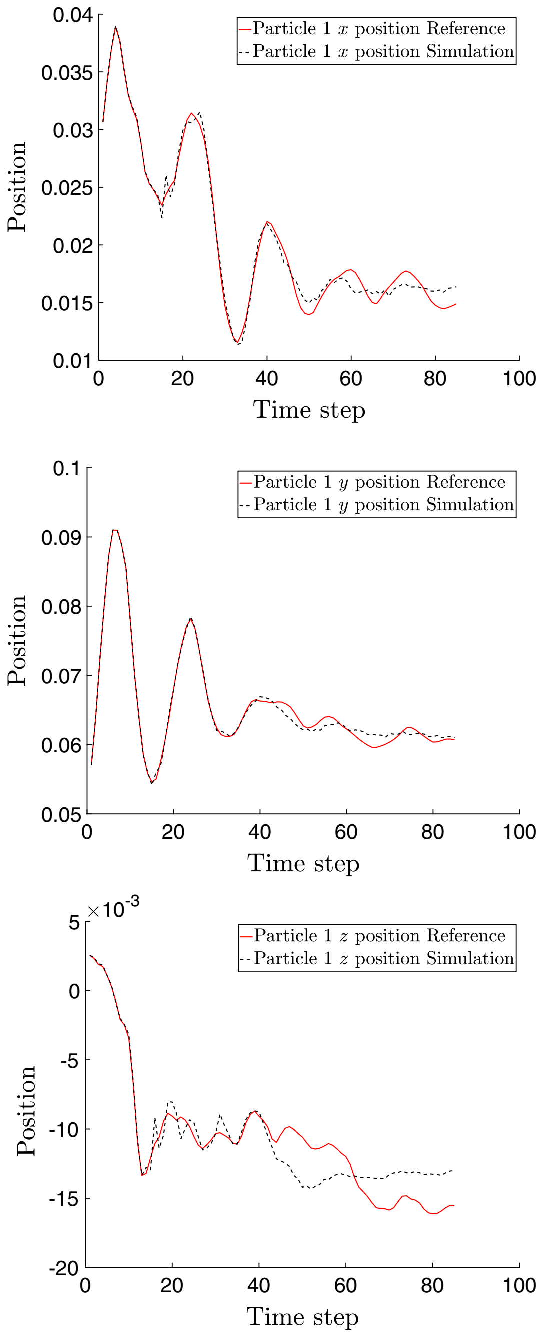

Fig. 6 POD-based reconstruction of the trajectory of one particular particle (evolution of coordinates x, y and z along time) vs. pseudo-experimental results. Initial velocity 0.15 m/s

The reconstruction of the displacement in time of a particular particle (number 1) is shown in Fig. 7.

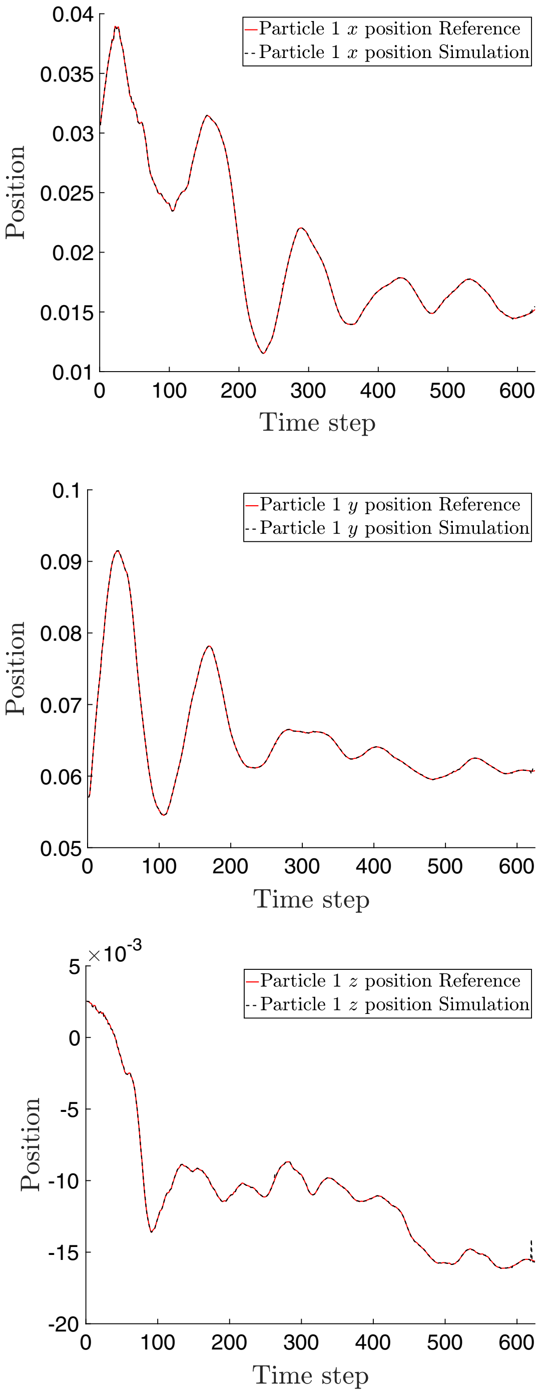

Fig. 7 LLE-based reconstruction of the trajectory of one particular particle (evolution of coordinates x, y and z along time) versus pseudo-experimental results. Initial velocity 0.15 m/s

4.3 Verification of the TDA approach

Similar to the LLE approach, TDA requires enough information to properly unveil the topological shape of data. For this reason, we pruned the data every $\Delta t = 0.00272$ s (i.e., we took one snapshot out of every seventeen in the SPH results). Each trajectory is successfully embedded in a manifold of $d = 3$ dimensions for calculating the set of $\{\mathsf{M}(\boldsymbol{\xi}_t), \boldsymbol{A}(\boldsymbol{\xi}_t), \boldsymbol{B}(\boldsymbol{\xi}_t)\}$ matrices.

Remark 1 The set of constituents of the GENERIC expression of the problem can not, in general, be obtained in the physical space $\mathcal{M} \subset \mathbb{R}^D$ and then projected to the reduced-order, embedded manifold. We risk to loose the properties of these terms so as to ensure thermodynamic consistency. Instead, they are obtained in the embedded space $\mathbb{R}^d$, after the necessary embedding of the data. Therefore, the regression introduced in Eq. (8) is actually accomplished as

$$\boldsymbol{\mu}^* = \{\mathsf{M}, \boldsymbol{A}, \boldsymbol{B}\} = \arg\min_{\boldsymbol{\mu}} ||\boldsymbol{\xi}(\boldsymbol{\mu}) - \boldsymbol{\xi}^{\texttt{meas}}||, \tag{9}$$

where the precise form of L is assumed known, as already mentioned.

The L_2-norm error in the reconstruction of the velocity field for this problem was 0.00011%, substantially lower than that of the POD and LLE approaches, see Fig. 8. This very low error justifies the selection of the TDA-based approach for a general case, whose details will be analyzed next.

4.4 Verification of the proposed integration scheme

Of course, the final goal of the just developed method is not to reproduce one of the experimental results—something that has been done just for verification of the approach—but to be able to integrate an arbitrary trajectory in the manifold described by the experimental results.

To test the efficacy of the integration scheme, we are going to compare the results calculated by this method with pseudo-experimental data from a trajectory other than the pseudo-experimental data employed to describe the manifold. From Eq. (7)—recall also Fig. 1—we obtain a scheme of the form

$$\boldsymbol{\xi}_{n+1} = \boldsymbol{\xi}_n + \Delta t \left[\mathsf{L}(\boldsymbol{\xi}_n)\mathbf{A}(\boldsymbol{\xi}_n) + \mathsf{M}(\boldsymbol{\xi}_n)\mathbf{B}(\boldsymbol{\xi}_n)\right]\boldsymbol{\xi}_n, \tag{10}$$

where we have highlighted the dependence of every term of the GENERIC description of the movement on $\boldsymbol{\xi}_n$, if an explicit scheme is chosen. Since no pseudo-experimental results for $\boldsymbol{\xi}_n$ will exist, in general, these values must be interpolated on the manifold by leveraging its just found geometrical structure.

We have performed the integration of a new trajectory with initial velocity $v_0 = 0.0175$ m/s, and compared the results obtained by the GENERIC integrator and those that we have obtained by SPH methods. A comparison of selected snapshots is shown in Fig. 9. Note the visible similarity between the results obtained by the proposed method and those obtained by SPH.

In Fig. 10 we show the reconstruction of the displacement of particle number 1. Of course, the error increases with respect to the examples in the preceding sections, since there is no experimental result that coincides with the simulated problem. The time evolution of the relative error in the prediction of the height reached by the liquid is plotted in Fig. 11. It is worth noting that the error remains around 2% in the vast majority of the time increments. Only in a small portion of the increments the error rise to a value always under 7%. Of course, further refinement in the data sampling will produce more accurate results.

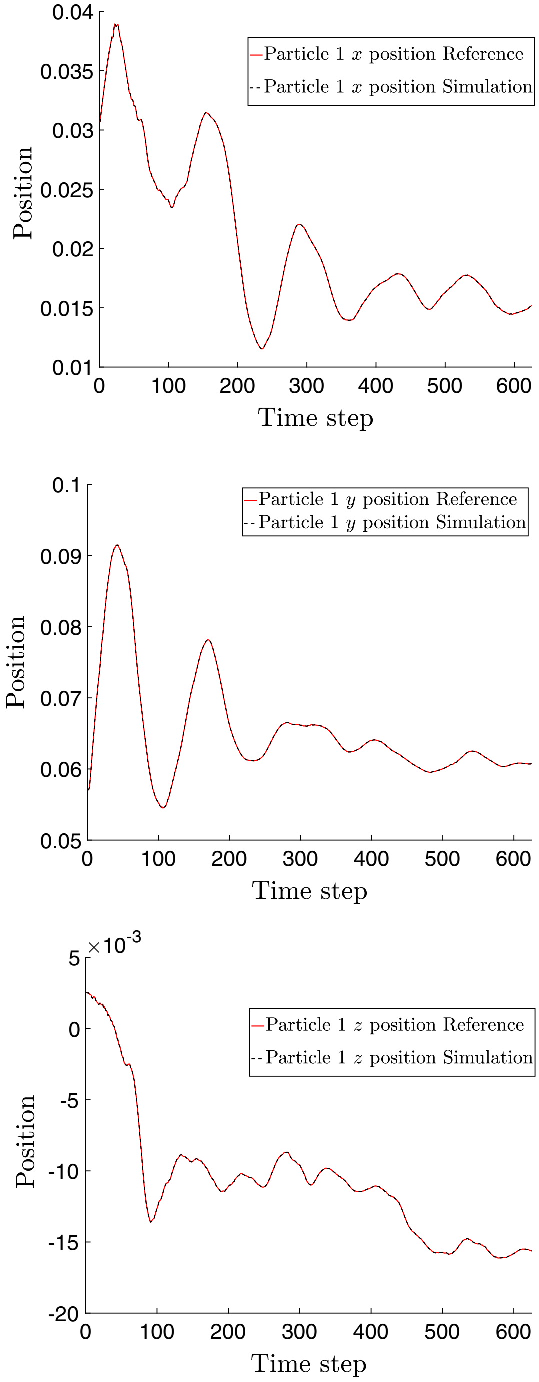

Fig. 8 TDA-based reconstruction of the trajectory of one particular particle (evolution of coordinates x, y and z along time) vs. pseudo-experimental results. Initial velocity 0.15 m/s

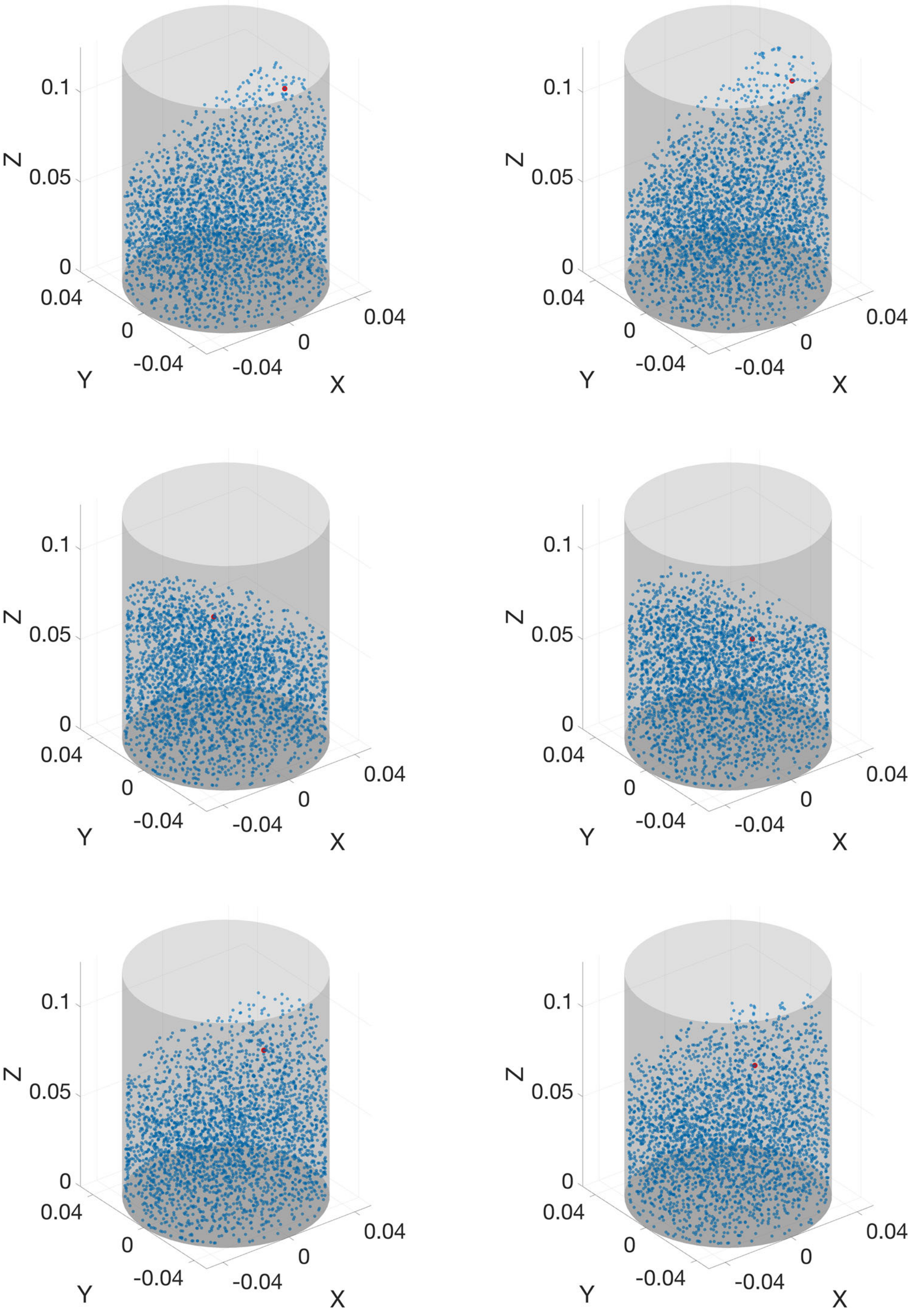

Fig. 9 Comparison between snapshots obtained by the proposed method (left column) and their equivalent ground truth obtained by SPH (right). Time instants 28, 115 and 172 are shown. Particle 1 is highlighted in red so as to ease the comparison among pseudo-experimental and learned results

The method runs faster than real time. For instance, a simulation of 1.7 s of physical time takes 1.640434 s running Matlab on a 2015 MacBook Pro laptop equipped with an Intel Core i7 processor. Of these, 0.901420 s correspond to the integration in time of the GENERIC expression. The remaining time is related to the neighbor finding process in the manifold of the results.

5 Discussion

What we have developed in this work is a practical way of learning the behavior of a free surface fluid that allows to overcome the need to integrate in time the Navier–Stokes equations, whose difficulty is well known. The method begins by writing down the equations of the fluid in the most general framework that allows for a description of the dynamics of

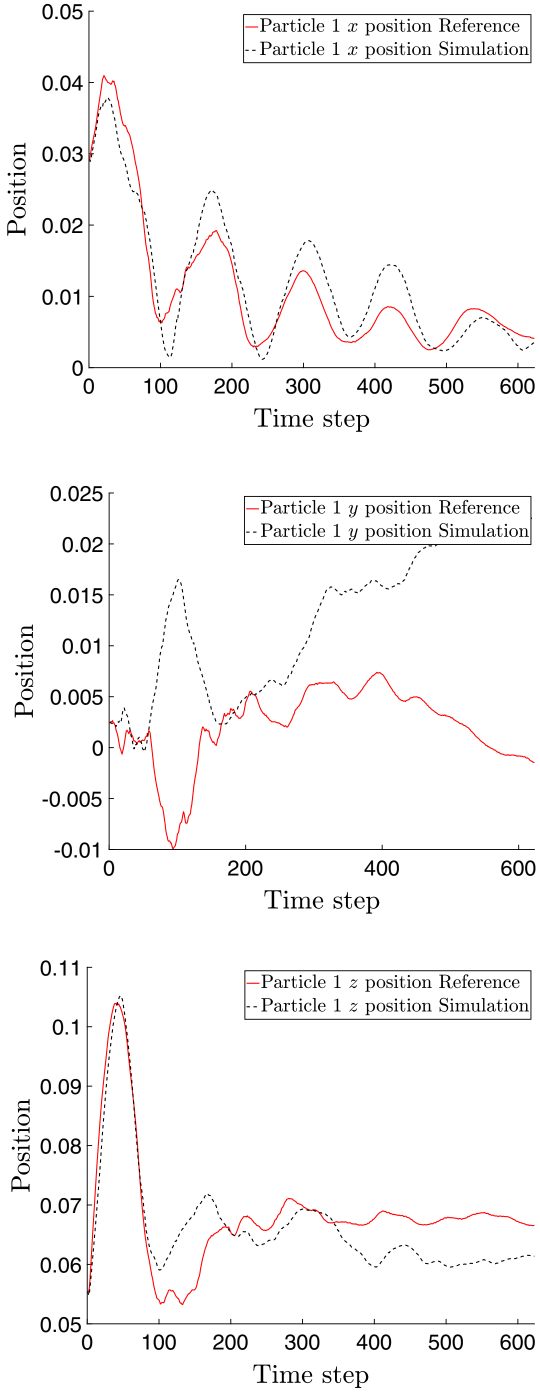

Fig. 10 Integration of the sloshing movement versus ground truth. Initial velocity 0.175 m/s. Figures represent, respectively from top to bottom, the displacement of particle number 1 along x, y and z coordinates

the fluid without losing information on the displacement and velocity fields. This is the so-called GENERIC framework.

GENERIC has a number of appealing features. First, its terms can be obtained numerically by regression of available experimental data. This approach guarantees the correct satisfaction of the conservation of energy and dissipation of entropy. Second, numerical discretization of GENERIC by finite differences in time provides us with a powerful and consistent integration scheme that has show to posses remarkable numerical properties, as previously studied by different authors.

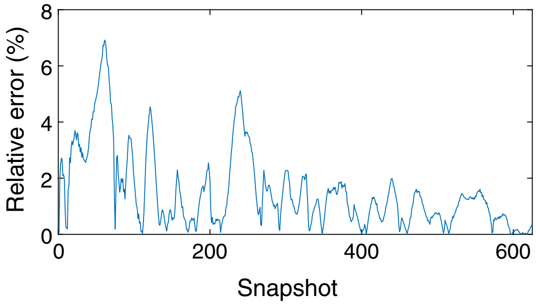

Fig. 11 Evolution in time of the error in the prediction of the water heigth

The third ingredient of the method is nonlinear model order reduction. Since GENERIC establishes a system of equations por every particle in the models, and given the fact that this number is usually very high, we have decided to reduce the order of the model by employing TDA-informed locally linear embeddings, thus greatly minimizing the number of degrees of freedom.

The combination of these ingredients has allowed us to develop a method that is able to cope with severe real time constraints, yet maintaining its desired thermodynamic consistency. Results showed that the simulations ran slightly faster than real time, with remarkable error levels always under 10%. This level of error is considered enough for many applications such as computer games, rendering, robot control, etc.

Acknowledgements This work has been supported by the Regional Government of Aragon and the European Social Fund, research group T24 17R. The support given by the ESI Group to F. Ch. through the Chair at ENSAM ParisTech, as well as the funding provided to E.C., D.G. and I.A. through the project "Simulated Reality: An intelligence augmentation system based on Hybrid Twins and Augmented Reality" is gratefully acknowledged.

References

1. Alla A, Kutz JN (2017) Nonlinear model order reduction via dynamic mode decomposition. SIAM J Sci Comput 39(5):B778–B796
2. Bender J, Koschier D (2015) Divergence-free smoothed particle hydrodynamics. In: Proceedings of the 14th ACM SIGGRAPH/eurographics symposium on computer animation, ACM. pp 147–155
3. Bessa MA, Bostanabad R, Liu Z, Hu A, Apley DW, Brinson C, Chen W, Liu WK (2017) A framework for data-driven analysis of

materials under uncertainty: countering the curse of dimensionality. Comput Methods Appl Mech Eng 320:633–667

4. Breiman L (2001) Random forests. Mach Learn 45(1):5–32
5. Brunton SL, Proctor JL, Kutz JN (2016) Discovering governing equations from data by sparse identification of nonlinear dynamical systems. Proc Natl Acad Sci 113:3932–3937
6. Carlsson G, Ishkhanov T, de Silva V, Zomorodian A (2008) On the local behavior of spaces of natural images. Int J Comput Vis 76(1):1–12
7. Epstein M (2010) The geometrical language of continuum mechanics. Cambridge University Press, Cambridge
8. Español P (2004) Statistical mechanics of coarse-graining. Springer, Berlin, pp 69–115
9. Gingold R, Monahan JJ (1977) Smoothed particle hydrodynamics: theory and applications to non-spherical stars. Mon Not R Astron Soc 181:375–389
10. González D, Chinesta F, Cueto E (2018) Thermodynamically consistent data-driven computational mechanics. Contin Mech Thermodyn. https://doi.org/10.1007/s00161-018-0677-z
11. González D, Chinesta F, Cueto E (2019) Learning corrections for hyperelastic models from data. Front Mater 6:14
12. Gremela M, Oettinger HC (1997) Dynamics and thermodynamics of complex fluids. I. Development of a general formalism. Phys Rev E 56:6620–6632
13. Huerta A, Liu WK (1988) Viscous flow with large free surface motion. Comput Methods Appl Mech Eng 69(3):277–324
14. Ibañez R, Abisset-Chavanne E, Gonzalez D, Duval JL, Cueto E, Chinesta F (2018) Hybrid constitutive modeling: Data-driven learning of corrections to plasticity models. Int J Mater Form
15. Ibanez R, Abisset-Chavanne E, Aguado JV, Gonzalez D, Cueto E, Chinesta F (2018) A manifold learning approach to data-driven computational elasticity and inelasticity. Arch Comput Methods Eng 25(1):47–57
16. Ibáñez R, Abisset-Chavanne E, Ammar A, González D, Cueto E, Huerta A, Duval JL, Chinesta F (2018) A multidimensional data-driven sparse identification technique: the sparse proper generalized decomposition. Complexity 2018:5608286. https://doi.org/10.1155/2018/5608286
17. Ibañez R, Borzacchiello D, Aguado JV, Abisset-Chavanne E, Cueto E, Ladeveze P, Chinesta F (2017) Data-driven non-linear elasticity: constitutive manifold construction and problem discretization. Comput Mech 60(5):813–826
18. Kaiser E, Nathan KJ, Brunton SL (2018) Discovering conservation laws from data for control. arXiv preprint arXiv:1811.00961
19. Kevrekidis Y, Samaey G (2010) Equation-free modeling. Scholarpedia 5(9):4847
20. Kim B, Azevedo VC, Thuerey N, Kim T, Gross MH, Solenthaler B (2018) Deep fluids: a generative network for parameterized fluid simulations. CoRR. arxiv:abs/1806.02071
21. Kirchdoerfer T, Ortiz M (2016) Data-driven computational mechanics. Comput Methods Appl Mech Eng 304:81–101
22. Kutz JN, Brunton SL, Brunton BW, Proctor JL (2016) Dynamic mode decomposition: data-driven modeling of complex systems. SIAM
23. L'ubor L, SoHyeon J, Barbara S, Marc P, Markus G (2015) Data-driven fluid simulations using regression forests. ACM Trans Gr 34(6):199:1–199:9
24. Lam RR, Horesh L, Avron H, Willcox KE (2017) Should you derive, or let the data drive? an optimization framework for hybrid first-principles data-driven modeling. *arXiv preprint*arXiv:1711.04374
25. Ly HV, Tran HT (2001) Modeling and control of physical processes using proper orthogonal decomposition. Math Comput Model 33(1–3):223–236
26. Millán D, Arroyo M (2013) Nonlinear manifold learning for model reduction in finite elastodynamics. Comput Methods Appl Mech Eng 261–262:118–131
27. Müller M, Charypar D (2003) Gross Markus particle-based fluid simulation for interactive applications. In: Proceedings of the 2003 ACM SIGGRAPH/eurographics symposium on computer animation, SCA '03, Aire-la-Ville, Switzerland, Switzerland. Eurographics Association, pp 154–159
28. Munch E (2017) A user's guide to topological data analysis. J Learn Anal 4(2):47–61
29. Öttinger HC (2005) Beyond equilibrium thermodynamics. Wiley, Hoboken
30. Pavelka M, Klika V, Grmela M (2018) Multiscale thermodynamics. De Gruyter, Berlin
31. Peherstorfer B, Willcox K (2016) Data-driven operator inference for nonintrusive projection-based model reduction. Comput Methods Appl Mech Eng 306:196–215
32. Proctor JL, Brunton SL, Nathan Kutz J (2016) Dynamic mode decomposition with control. SIAM J Appl Dyn Syst 15(1):142–161
33. Proctor JL, Brunton SL, Nathan Kutz J (2018) Generalizing koopman theory to allow for inputs and control. SIAM J Appl Dyn Syst 17(1):909–930
34. Romero I (2009) Thermodynamically consistent time-stepping algorithms for non-linear thermomechanical systems. Int J Numer Methods Eng 79(6):706–732
35. Romero I (2010) Algorithms for coupled problems that preserve symmetries and the laws of thermodynamics: Part i: Monolithic integrators and their application to finite strain thermoelasticity. Comput Methods Appl Mech Eng 199(25–28):1841–1858
36. Romero I (2013) A characterization of conserved quantities in non-equilibrium thermodynamics. Entropy 15(12):5580–5596
37. Roweis ST, Saul LK (2000) Nonlinear dimensionality reduction by locally linear embedding. Science 290(5500):2323–2326
38. Schenck C, Fox D (2017) Reasoning about liquids via closed-loop simulation. arXiv preprint arXiv:1703.01656
39. Schenck C, Fox D (2018) Perceiving and reasoning about liquids using fully convolutional networks. Int J Robot Res 37(4–5):452–471
40. Schenck C, Fox D (2018) Spnets: differentiable fluid dynamics for deep neural networks. arXiv preprint arXiv:1806.06094
41. Vázquez-Quesada A, Ellero M, Español P (2009) Consistent scaling of thermal fluctuations in smoothed dissipative particle dynamics. J Chem Phys 130(3):034901
42. Wasserman L (2018) Topological data analysis. Ann Rev Stat Appl 5(1):501–532
43. Williams CKI (1998) Prediction with Gaussian processes: from linear regression to linear prediction and beyond. Springer, Dordrecht, pp 599–621