Beatriz Moya Garcia

Learned simulation as the engine of physical scene understanding

Director/es

Cueto Prendes, Elías
González Ibáñez, David

ISSN 2254-7606

Prensas de la Universidad
Universidad Zaragoza

Servicio de Publicaciones

ISSN 2254-7606

Tesis Doctoral

LEARNED SIMULATION AS THE ENGINE OF PHYSICAL SCENE UNDERSTANDING

Autor

Beatriz Moya Garcia

Director/es

Cueto Prendes, Elías
González Ibáñez, David

UNIVERSIDAD DE ZARAGOZA
Escuela de Doctorado

Programa de Doctorado en Ingeniería Mecánica

2022

Tesis Doctoral

Learned simulation as the engine of physical scene understanding

Autor

Beatriz Moya García

Director/es

David González Ibáñez
Elías Cueto Prendes

UNIVERSIDAD DE ZARAGOZA

2022

Learned simulation as the engine of physical scene understanding

PhD thesis by
Beatriz Moya

Doctoral Advisors
Prof. David González
Prof. Elías Cueto

PhD Programme in Mechanical Engineering
April 2022

Universidad
Zaragoza
1542

Learned simulation as the engine of physical scene understanding

PhD thesis by

Beatriz Moya

Doctoral advisors

Prof. David González
Prof. Elías Cueto

PhD Programme in **Mechanical Engineering**
April 2022

Universidad
Zaragoza
1542

A mis padres
A Javi
A mi abuelo

Agradecimientos

Creo que David González, Icíar Alfaro y Elías Cueto se merecen un lugar de honor en estos agradecimientos, y por eso tienen que ser los primeros. No sé qué será de mi en un futuro, pero lo que sí que sé es que si he disfrutado mi tesis ha sido gracias a vosotros. Transformasteis mis dudas en un comienzo y, cuatro años después, creo que compartir este tiempo con vosotros ha sido una de las mejores decisiones que he tomado. Gracias por enseñarme todo lo que sé, y por estar ahí en los momentos difíciles. También querría dar las gracias a Alberto, por sus ideas y sus enseñanzas, y a Quercus, por su seguridad y sus dibujos en pizarra. Carlos, espero que podamos seguir trabajando mucho juntos. Como ves, este es un gran grupo.

Tampoco habría sido lo mismo sin mis compañeros de sala. La familia va creciendo poco a poco, y es precioso veros avanzar como investigadores y personas. Espero que me permitáis que os mande un abrazo de grupo que resuma lo feliz que he sido con todos vosotros. Entre ellos querría destacar a Patri, Itzíar, Iulen, Pau, Javi y Jose. Algunos pasaron como un rayo, como Julio, y con otros, como Abel, tuvimos que coincidir de manera virtual. Sea como sea, os habéis hecho un hueco en mi vida para siempre.

Si hablo del grupo de doctorandos no me puedo dejar al resto de profesores del área, así como al equipo del I3A. Sois la maquinaria que nos permite movernos hacia adelante.

A lo largo del camino me he encontrado con personas inspiradoras. Gracias Prof. Paco Chinesta y Prof. Miroslav Grmela, por vuestras ideas, porque sin ellas la tesis no habría sido la misma.

Me siento también muy afortunada de haber podido participar en congresos divulgando mi trabajo, un privilegio en estos años. Ahí conocí a Jonatha, un buen amigo a pesar la distancia.

También querría agradecer al Prof. Jan S. Hesthaven haberme dado la oportunidad de hacer una estancia de investigación con él. Han sido seis meses de aprendizaje fabulosos que no habrían sido lo mismo sin Junming, Ricardo, Mariella, Niccolò, Nicolò y Federico. Os deseo de corazón todo el éxito que merecéis.

Pienso en mi familia, la suerte de haberla tenido siempre junto a mí, y en estos años que hemos pasado. Si hemos sobrevivido ha sido por el amor que nos tenemos. Gracias a todos por apoyarme, a los que llevan conmigo desde que nací, y los que me han recibido como un miembro más.

Si hablo de la familia que recibí, no puedo dejar de hablar de la que me ha dado la vida. Mis amigas, o mis hermanas, que a veces no entienden lo que hago, pero me apoyan de manera incondicional.

A mis padres, que me han dado todas las herramientas para crear oportunidades y disfrutar de la vida y de mi educación. Dice mi padre que, si me dejan algo, que sea una vida plena, y puedo afirmar con seguridad que me la han dado. Gracias por estar conmigo, darme alas, y creer siempre en mí y mi futuro.

A Javi. Agradezco haber podido compartir este tiempo contigo, y también espero compartir mi futuro y una vida feliz. Has sido mi paz y mis fuerzas cuando ambas flaqueaban, pero, sobretodo, mi felicidad.

Finalmente agradezco a la Universidad de Zaragoza, al Ministerio de Ciencia e Innovación, y al programa de becas Cai-Ibercaja por haber puesto en valor mi trabajo.

Y a ti, a mi abuelo, por hacerme la mujer que soy y ser parte de mí. Siempre has estado ahí, desde mi primer día de colegio hasta que se te agotaron las fuerzas. Ojalá pudieras verme ahora, aunque estoy segura de que, de alguna manera, me acompañas. Siempre te llevaré en mi corazón.

Acknowledgements

I believe David González, Icíar Alfaro, and Elías Cueto deserve an honored place in this section, and that's why they will be the first. I don't know what will happen to me in the future, but I know for sure that I enjoyed the path to this moment thanks to you. You made my doubts disappear to start a Ph.D. and, four years later, I think that sharing this part of my life with you was one of the best decisions that I've ever made. Thank you for teaching me all I know, and being there for me in the hard times. I would also like to thank Alberto, for his ideas and lessons, and Quercus, for his confidence and drawings on the board. Carlos, I hope we can keep working together. As you can see, this is a great group.

This experience wouldn't have been the same without our group of doctoral students. This family keeps growing, and it is wonderful to see you become great researchers. Allow me to send you a big hug, to summarize how happy I was for sharing this time with you. Among them, I would to acknowledge particularly Patri, Itzíar, Iulen, Pau, Javi, and Jose for being there. Like Julio, some went by like a clap of thunder, and we had to meet virtually with others like Abel. No matter how it was, you will always have a place in my heart.

If I talk about the group of doctorates, I cannot forget the professors of the department and the I3A team. You are the machinery that pushes us forward.

Along the way, I met inspiring people. Thank you professors Prof. Paco Chinesta and Prof. Miroslav Grmela for your ideas, because without them this dissertation wouldn't have been the same.

I'm also thankful for the privilege of attending congresses, where I met Jonatha, a good friend in the distance.

I would also like to thank Prof. Jan S. Hesthaven for giving me the opportunity to, in these times, do a research stay at EPFL. During these six months, I tried to learn as much as possible, and this experience would not have been the same without Junming, Ricardo, Mariella, Niccolo, Nicollo, and Federico. I wish you all the good luck and success that you deserve.

I think of my family, and the years we passed through. We survived together thanks to the love we have for each other. Thanks for your support, to the family that met me when I was born, and to the family that warmly welcomed me.

If I talk about my family, I have to talk about the one I was given in life. My friends, my siblings, who do not always understand what I do, but always offer their unconditional support.

To my parents, who gave me the tools to create opportunities and enjoy my life and education. Instead of inheriting a fortune, my father says that they want to live a full life with me, and I can ensure you I did. Thanks for being there with me, for the wings, and for believing in me and my future.

To Javi. You cannot imagine how grateful I am for having shared this time with you, and I hope to build a future together. You were my tranquillity and strength when I was running out of them but, above all, my happiness.

Finally, I am thankful for the support of the University of Zaragoza, the Ministry of Science, and the funding program Cai-Ibercaja for the value that you gave to my work.

And to you, my grandfather, for making me who I am and being part of me. You have always been there, from my first day of school until your strength gave out. I hope you could see me now, but I am sure you are with me in some way. You will always be in my heart.

All our knowledge begins with the senses, proceeds then to the understanding and ends with reason. There is nothing higher than reason.

Immanuel Kant
Critique of Pure Reason

Abstract

Cognition evokes human abilities for reasoning, communication, and interaction. This includes the interpretation of real-world physics so as to understand its underlying laws. Theories postulate the similarity of human reasoning about these phenomena with simulations for physical scene understanding, which gathers perception for comprehension of the current dynamical state, and reasoning for time evolution prediction of a given system.

In this context, we propose the development of a system for learned simulation. Given a design objective, an algorithm is trained to learn an approximation to the real dynamics to build a digital twin of the environment. Then, the underlying physics will be emulated with information coming from observations of the scene. For this purpose, we use a commodity camera to acquire data exclusively from video recordings.

We focus on the sloshing problem as a benchmark. Fluids are widely present in several daily actions and portray a physically rich challenge for the proposed systems. They are highly deformable, nonlinear, and present a dominant dissipative behavior, making them a complex entity to be emulated. In addition, we only have access to partial measurements of their dynamical state, since a commodity camera only provides information about the free surface.

The result is a system capable of perceiving and reasoning about the dynamics of the fluid. This cognitive digital twin provides an interpretation of the state of the fluid to integrate its dynamical evolution in real-time, updated with information observed from the real twin. The system, trained originally for one liquid, will be able to adapt itself to any other fluid through reinforcement learning and produce accurate results for previously unseen liquids. Augmented reality is used in the design of this application to offer a visual interpretation of the solutions to the user, and include information about the dynamics that is not accessible to the human eye.

This objective is to be achieved through the use of manifold learning and machine learning techniques, such as neural networks, enriched with physics information. We use inductive biases based on the knowledge of thermodynamics to develop machine intelligence systems that fulfill these principles to provide meaningful solutions to the dynamics.

This problem is considered one of the main targets in fluid manipulation for the development of robotic systems. Pursuing actions such as pouring or moving, sloshing dynamics play a capital role for the correct performance of aiding systems for the elderly or industrial applications that involve liquids.

Resumen

La cognición humana evoca las habilidades del razonamiento, la comunicación y la interacción. Esto incluye la interpretación de la física del mundo real para comprender las leyes que subyacen en ella. Algunas teorías postulan la semejanza entre esta capacidad de razonamiento con simulaciones para interpretar la física de la escena, que abarca la percepción para la comprensión del estado físico actual, y el razonamiento acerca de la evolución temporal de un sistema dado.

En este contexto se propone el desarrollo de un sistema para realizar simulación aprendida. Establecido un objetivo, el algoritmo se entrena para aprender una aproximación de la dinámica real, para construir así un gemelo digital del entorno. Entonces, el sistema de simulación emulará la física subyacente con información obtenida mediante observaciones de la escena. Para ello, se empleará una cámara estéreo para adquirir datos a partir de secuencias de video.

El trabajo se centra los fenómenos oscilatorios de fluidos. Los fluidos están presentes en muchas de nuestras acciones diarias y constituyen un reto físico para el sistema propuesto. Son deformables, no lineales, y presentan un carácter disipativo dominante, lo que los convierte en un sistema complejo para ser aprendido. Además, sólo se tiene acceso a mediciones parciales de su estado ya que la cámara sólo proporciona información acerca de la superficie libre.

El resultado es un sistema capaz de percibir y razonar sobre la dinámica del fluido. El gemelo digital cognitivo así construido proporciona una interpretación del estado del mismo para integrar su evolución en tiempo real, aprendiendo con información observada del gemelo físico. El sistema, entrenado originalmente para un líquido concreto, se adaptará a cualquier otro a través del aprendizaje por refuerzo produciendo así resultados precisos para líquidos desconocidos.

Finalmente, se emplea la realidad aumentada (RA) para ofrecer una representación visual de los resultados, así como información adicional sobre el estado del líquido que no es accesible al ojo humano.

Este objetivo se alcanza mediante el uso de técnicas de aprendizaje de variedades, y aprendizaje automático, como las redes neuronales, enriquecido con información física. Empleamos sesgos inductivos basados en el conocimiento de la termodinámica para desarrollar un sistema inteligente que cumpla con estos principios para dar soluciones con sentido sobre la dinámica.

El problema abordado en esta tesis constituye una dificultad de primer orden en el desarrollo de sistemas robóticos destinados a la manipulación de fluidos. En acciones como el vertido o el movimiento, la oscilación de los líquidos juega un papel importante en el desarrollo de sistemas de asistencia a personas con movilidad reducida o aplicaciones industriales.

Contents

Contents

Thesis 1

1 Introduction 3

1.1 General overview . . . 3
1.2 Simulation for physical scene understanding . . . 4
1.3 Model problem . . . 5
1.4 Problem description . . . 8
1.4.1 Data acquisition . . . 8
1.4.2 Real-time interaction . . . 9
1.4.3 Physics predictions with learned simulators . . . 10
1.4.4 Closed loop simulation and real-world interaction . . . 12
1.4.5 Augmented reality representation . . . 13
1.5 Data-driven computational mechanics . . . 14
1.5.1 Classical Regression . . . 16
1.5.2 Neural Network approaches . . . 17
1.5.3 Machine Learning for Fluid Dynamics . . . 18
1.6 Objectives . . . 19
1.7 Thesis structure . . . 20

2 A first attempt based on manifold learning 23

2.1 Introduction . . . 23
2.2 The dynamical systems equivalence . . . 24
2.3 Data collection . . . 26
2.4 Manifold learning techniques . . . 28
2.4.1 Proper Orthogonal Decomposition (POD) . . . 28
2.4.2 Locally Linear Embedding (LLE) . . . 30
2.4.3 Topological Data Analysis (TDA) . . . 31

2.5 Dynamical systems with known properties 33
2.5.1 GENERIC 34
2.6 Numerical results 36
2.6.1 Manifold learning evaluation 37
2.6.2 Integration in the reduced-order manifold 42
2.7 Conclusions 43

3 Developing physically sound, self-learning digital twins for sloshing fluids 47
3.1 Introduction 47
3.2 Learning Newtonian and non-Newtonian fluids 49
3.3 Manifold learning of fluids: the k-PCA 52
3.4 Dynamics reconstruction from acquired data 53
3.5 Fluid clustering and classification 54
3.6 Observation: the camera model 55
3.7 Data extraction from the scene 59
3.7.1 Feature detection and tracking 59
3.7.2 Simulation in the loop and twin representation 60
3.7.3 Detection and data acquisition of the free surface 61
3.8 Experimental validation of the twin 61
3.9 Conclusions 63

4 On the use of thermodynamics as an inductive bias for neural networks 67
4.1 Introduction 67
4.2 Problem description 69
4.3 Artificial Neural Network Theory 70
4.4 Method 71
4.4.1 Model order reduction with autoencoders 73
4.4.2 Structure-preserving neural networks 75
4.4.3 Recurrent neural networks to recover dynamical information . . . 76
4.5 Computational training and validation 78
4.5.1 Hyperparameters and training details 78
4.5.2 Computational validation 80
4.6 Coupling with the real liquid twin 81
4.6.1 Computer vision 83
4.6.2 Results 84
4.7 Conclusions 88

5 Perceiving and reasoning about previously unseen scenarios 91

5.1 Introduction . . . 91

5.2 Method . . . 93

5.2.1 Reinforcement learning background . . . 95

5.2.2 Model adaptation from observation . . . 96

5.3 Results . . . 98

5.3.1 Computational validation . . . 98

5.3.2 Testing in real scenarios . . . 102

5.4 Conclusions . . . 107

6 Conclusions 111

6.1 Introduction . . . 111

6.2 General conclusions . . . 111

6.3 Thesis contributions . . . 114

6.4 Future research lines . . . 115

6.4.1 GENERIC into learning . . . 115

6.4.2 Fluid modeling . . . 115

6.4.3 Virtual interaction and representation . . . 116

6.5 Publications and contributions . . . 116

6.5.1 Journal contributions . . . 116

6.5.2 Book chapters . . . 117

6.5.3 Oral communications in conferences and workshops . . . 117

6.5.4 Poster communications in conferences and workshops . . . 118

6.5.5 Symposium organization . . . 118

6.5.6 Outreach communications . . . 119

6.5.7 Free software . . . 119

6.6 Research stay . . . 119

A Conclusiones 121

A.1 Introducción . . . 121

A.2 Conclusiones . . . 121

A.3 Contribuciones de la tesis . . . 124

A.4 Líneas de trabajo futuras . . . 125

A.4.1 GENERIC en el aprendizaje . . . 125

A.4.2 Modelado de fluidos . . . 126

A.4.3 Interacción virtual y representación . . . 126

A.5 Publicaciones y contribuciones 127
A.5.1 Publicaciones en revista científica 127
A.5.2 Capítulos de libro 127
A.5.3 Participación en congresos 128
A.5.4 Participaciones con póster en conferencias y workshops 128
A.5.5 Organización de sesiones en congresos 129
A.5.6 Diseminación general 129
A.5.7 Software libre 129
A.6 Estancias de doctorado 130

B Artículo de divulgación: Aprender a razonar, el ser o no ser de la robótica 133

C Figments of reality 137

D Abbreviations 139

Bibliography 141

List of Figures

1.1 Learning cycle. Visual inputs are perceived and data is converted into information for reasoning. At this stage, learned physical models inferred from experience are used for the prediction of the future state and decision-making in critical situations. z_t represents the information available about the dynamics. In contrast, s_t represents the full state of the dynamics at a specific time step 5

1.2 Examples of slosh. (A) Baby learning fluid dynamics through interaction. In manipulation, the slosh is one of the critical concerns to pursue a task. (B) Examples of study of the slosh in robotic manipulation to mimic human capacities [Schenck and Fox, 2018a] and analysis of wall forces in tanks undergoing sloshing [Kim et al., 2004]. (C) Visualization of sloshing dynamics through augmented reality to enrich the real environment [Fujisawa and Kato, 2009] (D) Approximation of the sloshing dynamics with the smooth particle hydrodynamics theory to model the perception system [Moya et al., 2019]. 7

1.3 Data-driven learning scale, adapted from [Karniadakis et al., 2021]. Methods are conditioned by the amount of data available and the physics included in the learning algorithm. Traditional dynamical models (i.e. Navier Stokes) require a little data for fitting, while big data scenarios usually employ purely data-driven algorithms. Physics-informed data-driven methods find a balance between the data provided, especially when we work in small data regimes, combined with physics to guide learning. 12

1.4 Reality-virtuality scale. It shows the progressive addition of virtual elements to the real setting. Augmented reality refers to the addition of virtual elements to reality, while augmented virtuality includes information from the real world in the virtual setting. Mixed reality operates in this range to perform the interaction between both realities. 13

1.5 Inductive bias classification for the different types of knowledge that can be included into learning described in [Karniadakis et al., 2021]. 16

1.6 Thesis outline. The development of the perception algorithm is broken down into four steps. A chapter is dedicated to each of them. 21

2.1 Geometry of the liquid and the glass. The problem is defined for one geometry and one type of liquid. 25

2.2 Cloud of nodes of the discretization of the liquid at rest, which is its initial state for the simulation of all velocities. There are 2898 particles in the discretization. Particle with label 1, for which some results are analyzed hereafter, is highlighted in red. 27

2.3 Hypothesis about 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 $\mathbb{R}^D$. An arbitrary trajectory of the system in the phase space is represented in red. Dimensionality reduction is applied to project the data to an embedding space in $\mathbb{R}^d$, with $d \ll D$ where the system dynamics will be integrated. 29

2.4 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. 32

2.5 Evolution of the eigenvalues of the pseudo-experimental results in POD approach. 38

2.6 POD-based reconstruction of the trajectory of one particular particle (evolution of coordinates x, y and z along time) vs. pseudoexperimental results. Initial velocity 0.15 m/s. 39

2.7 LLE-based reconstruction of the trajectory of one particular particle (evolution of coordinates x, y and z along time) vs. pseudoexperimental results. Initial velocity 0.15 m/s. 40

2.8 TDA-based reconstruction of the trajectory of one particular particle (evolution of coordinates x, y and z along time) vs. pseudoexperimental results. Initial velocity 0.15 m/s. 41

2.9 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. . . 44

2.10 Integration of the sloshing movement versus ground truth. Initial velocity 0.175 m/s. Fig.s represent, respectively from top to bottom, the displacement of particle with label 1 along x, y and z coordinates. 45

2.11 Evolution in time of the error in the prediction of the water height. 46

3.1 Containers used for simulation, representation and comparison with the virtual phantom. The three glasses are identical. The middle glass will be virtually filled with the replica of the liquid volume. Since it is empty, the painted texture helps the cameras to track its movement. 49

3.2 Standard classification of the fluid families considered herein. In the case of Newtonian fluids, their properties are constant over time and show a linear response. Their flow index n is then set to 1, and their yield stress to $\tau_0 = 0$. In contrast, shear thickening fluids start flowing when the stimulus is greater than the yield stress $\tau_0 > 0$. For these fluids, $n > 1$. In this work, fluids whose behavior can be assimilated to shear-thinning. Fluids that incorporate some kind of plastic behavior need special treatment and have not been considered yet. 51

3.3 This picture represents the evolution of the eigenvalues for the first 10 k-PCA modes for the whole set of fluids. We distinguish three modes that stand out over the others. This fact justifies the reduction performed. As a result, we aim to provide a more manageable and efficient system for classification. 54

3.4 By employing k-PCA we reach a manifold of 3 dimensions where the different fluids, represented by one color each, remain clustered. 55

3.5 Scheme of the pinhole model. It represents the projection of the 3D world onto 2D images. 57

3.6 Bundle adjustment. Three cameras, whose rotation and translation information are unknown, capture the same features of an object to perform the optimization over the unknown data and perform the estimation of the cameras' position and 3D coordinates of the features of the object. . . 58

3.7 The picture shows the functioning of the stereo system in this work. We fix the location of the camera, which is referenced to the origin position through the extrinsic parameters. The two images provided by the stereo system output the desired 3D reconstruction of a point. The camera performs continuous triangulations and exports the depth of each pixel from the 2D matches detected between the right and left lenses. 59

3.8 Example of frame binarization. The picture is first transformed to gray scale for gentle binarization. Noise is also filtered to detect a smooth surface. . 62

3.9 Free surface detection and tracking in video sequence. The points selected to belong to the free surface are highlighted in red over the original frame for verification. 63

3.10 Representation of the quantifiable comparison of the real liquid and the replica. The free surface is defined as a function of its height at different points. These heights are compared in the same snapshot to evaluate the reconstruction error. 63

3.11 Snapshots employed for comparison between the real liquid and the digital twin. The free surface reconstruction has been evaluated to compute the error. 64

4.1 Graphic representation of the mathematical elements in ANNs. 71

4.2 Sketch of the construction procedure for the deep neural network. In the first step (first row), we perform model order reduction with autoencoders. Then, we train a structure-preserving neural network to integrate in time the state of the system. We do not have access to the high-dimensional state of the system, only a portion of it, in red in the input vector. The encoder is substituted by a RNN to find a mapping from the partial measurements to the low-dimensional manifold. 72

4.3 Graphical description of an autoencoder. The encoder learns a mapping to a latent space where the information of the input is compressed to a low dimensional representation. The decoder, usually symmetric to the encoder, undoes the process to map the hidden state to the full space. . . 74

4.4 Scheme of structure-preserving neural networks (SPNN). We introduce the vector of state variables $\boldsymbol{x}_n$, in the reduced space, into the network to learn L$_n$, M$_n$, DE$_n$, DS$_n$ and perform the time integration. The final output is the vector of state variables $\boldsymbol{x}_{n+1}$, in the reduced space. 76

4.5 Representation of a GRU cell. The three main paths indicated represent the update and reset gates, the new memory cell, and their connection to update the new hidden state transmitted to the next later. 78

4.6 Simulation results. Learning of the dynamics in the latent manifold. Dashed lines represent the time evolution of the latents that aimed to be emulated. Lines in blue represent the result of the SPNN in the latent manifold. 81

4.7 Time evolution of selected state variables evaluated at 21 random particles. The graph shows a comparison between the simulated fields with the ground truth for the validation simulation of the algorithm. 82

4.8 Comparison of the reconstruction of the integration provided by the SPNN (right) with the ground truth (left). The selected snapshots correspond to peaks of the sloshing dynamics of glycerine. Specifically, we present the comparison for snapshots 1, 33, and 64 of the collection. The height of the cup is 7 cm, and it is filled up to 5.6 cm approximately. 83

4.9 Color and depth stream before (up) and after (down) applying filters to reconstruct the depth map 85

4.10 Representation of the color frame and its conversion to a binarized image to seek the free surface. The area defined for searching is represented in the color frame as well as the points of the free surface detected in the black and white image. 85

4.11 Results for a 12 seconds video of a glass of glycerine. Eight snapshots of the sloshing sequence were selected for comparison. The selected snapshots have indexes 560, 565, 568, 572, 578 from left to right. The second row corresponds to the fluid reconstruction and prediction provided in the previous snapshot. From rows three to ten we show the additional information obtained from the reconstruction and simulation (velocity, energy, and stress fields, respectively). 86

4.12 Detail of the comparison of glycerine (left) and water (right) with the prediction. The third column of both liquids compares the predicted fluid volume (in blue), the free surface of the liquid volume (green) and the target free surface (in red). The RMSE and the Hausdorff distance (HD) that correspond to each snapshot are indicated. 87

4.13 Evolution of the mean squared error during the perception process of sloshing in a glass of glycerine. 88

4.14 Time derivatives of energy and entropy along the video. The time derivative of energy oscillates around zero, ensuring energy conservation. Entropy production is also ensured since the time derivative is always positive. 89

5.1 Original simulation engine trained for one liquid. It will be the starting point for the reinforcement learning algorithm. 94

5.2 Reinforcement learning scheme. Given the partial observation of the free surface $\boldsymbol{z}_n$ at time $t = n \times \Delta t$ our method estimates the full state of the fluid $\hat{\boldsymbol{s}}_{n+1}$. Its similarity with the next state $\boldsymbol{s}_{n+1}$ is evaluated to update the model. To preserve the patterns previously learnt, only a few layers are activated (in yellow) for backpropagation of the error and correction of the model. 96

5.3 Representation of the correction of the latent manifold of the position. The latent representation evolves to match the features of the new liquid. . . . 100

5.4 (Up) Relative error of the reconstruction of the free surface for the four simulations used for training and test. The relative error obtained with the source model is reduced after the optimization. (Down) Detail of the sloshing height reconstruction of the most critical simulation. The method correctly emulates the behavior in magnitude terms and, notably, in the precise time occurrence of the peaks. 101

5.5 Reconstruction before and after correction of water, blood and butter. . . 104

5.6 Representation of surface tracking. Each color frame is converted to a binary frame where the surface is detected (shown in red in the correspondent color frame). The depth map is built upon these points. For transparent liquids, like the glass of water shown in the picture, the depth map is of lower quality resulting in an incomplete detection of the surface. 105

5.7 Representation of the interpolation method for data acquisition. Given some detected points from the free surface, we interpolate the height at equally spaced locations to describe the free surface in an homogeneous reference to facilitate comparison and learning. 105

5.8 Relative error and reconstruction of the sloshing height for train and test recordings of water, beer, gazpacho and honey. 106

5.9 Reconstruction of state variables during time integration for beer. The reconstruction of the vertical velocity, energy and normal stress of the particles is shown at four integration snapshots 108

5.10 Render of the volume reconstruction before and after correction. The algorithm outputs a particle discretization of the fluid that can be presented as a three-dimensional render of the volume for visualization and interpretation. The figure shows the peaks of the dynamics observed in a piece of the recordings. 110

B.1 Representación del ciclo cognitivo para la interpretación de dinámica de fluidos 135

C.1 The research revolves around the development of living digital twins of fluid dynamics. The avatar of the fluid, discretized in particles, learns from reality and shows the results of the simulation with augmented reality, as we see in the picture. These twins are a reflection of a real fluid, like the reflection of the mirror. Even though they are not real, they act as if they were. Their behavior is portrayed in the picture by representing the particles as a recreation of the real fluid, and not the one mirrored. We don't have to find out the reality, because for machines this is the reality. 138

Thesis

Chapter 1

Introduction

1.1 General overview

The human brain has been an exceptional source of inspiration for science, research, and technology historically. Many disciplines have taken advantage of the enhancements resulting from recreating both the human cognitive system and neuron connections that learn the correlation of events. The brain has vast capabilities, some of them still unknown to us, but in the current decade, both questions and solutions arise exponentially concerning how we can exploit them.

Cognition is one of the brain's core functions. It encompasses the capacities related to knowledge acquisition, processing of information, reasoning, and decision making. It is thanks to them that a person can learn. In other words, these skills trigger our intelligence. Perception is one of the core skills of human cognition. This term refers to the interpretation of real events from information captured by our senses. Thus, physics perception is the interpretation of physical events.

A question may arise here: when do we learn this as we grow? Piaget's theory describes the four phases of intelligence development [Reynolds et al., 2007]. The first stage, known as *sensorimotor stage*, occurs between the ages of 0 and 2. One of the major developments in this step is the interpretation and knowledge internalization of the functioning of the world through the infants' movements and sensations. Toddlers learn basic actions, like grasping or, more importantly, listening and observing, to explore the behavior of elements that surround them. As a result, they learn that objects exist although they may be hidden (this is referred to as object permanence) and, more importantly, they understand that actions will have consequences in the environment thanks to their acquired reasoning capacities about real events.

In this context, physical scene understanding encompasses the comprehension of the scene and its link with the acquisition of knowledge and experience to forecast future events. [Battaglia et al., 2013] postulate that this ability can be represented as a simulation to understand real-world situations that involve physics.

The present work intends to contribute to the field of physical scene understanding by developing a system for performing learned simulations to mimic the behavior of dynamical systems from real observations focusing on the sloshing dynamics. It will provide an interpretation of the current dynamical state for reasoning about its dynamical evolution from observations of the real environment. The outcome is a digital twin of a real entity to mimic its dynamics as an interface with reality. It will interact with the environment like the real counterpart, and adapt to changes in the scenario [Liu and Negrut, 2021].

Some requirements for the development of these systems will be progressively addressed by enriching the proposed algorithm with new capabilities built upon computer vision, machine learning, and computational modeling. The method will be first tested on a common liquid: water. Then, the proposal will be extended to a wider variety of fluids with tools to distinguish the type of fluid in the implementation of a digital twin with computer vision systems. Based on this scheme, we will develop a means to learn from observations of the free surface. Finally, the algorithm will learn to adapt to previously unseen fluids based on reinforcement learning techniques.

1.2 Simulation for physical scene understanding

The problem presented in this thesis is denominated *physical scene understanding* [Battaglia et al., 2013]. This terminology encompasses systems capable of *seeing*, interacting, and reasoning around real dynamical events. In addition, there should be continuity in the learning to improve physics perception from experience without the imposition of new instructions.

The flow of information is represented in Fig. 1.1. Data is acquired by our senses, and then it is interpreted and converted into information. Then, simulation based on acquired knowledge provides an interpretation of the physics and prediction of future states. This information may be employed for decision-making if required. Knowledge will be continuously enriched from experience and new data perceived to improve predictions about reality.

Several works show diverse proposals for physical reasoning for solids and fluids [Mrowca et al., 2018] [Lieto et al., 2017] [Yildirim et al., 2015] [Schenck and Fox, 2016b] [Sanchez-Gonzalez et al., 2020] [Schenck and Fox, 2018a] [Schenck and Fox, 2018b]. These works explore knowledge and approximation of cognition based on simulators for prediction and decision making. [Battaglia et al., 2016] propose the so-called interaction networks to study dynamical systems. In this proposal, there are different algorithms trained for describing the object's behavior and correlations separately. The networks are graph-inspired to learn from evolving connections in the elements that interact to introduce the geometry of the dynamics as an inductive bias [Battaglia et al., 2018]. We call inductive

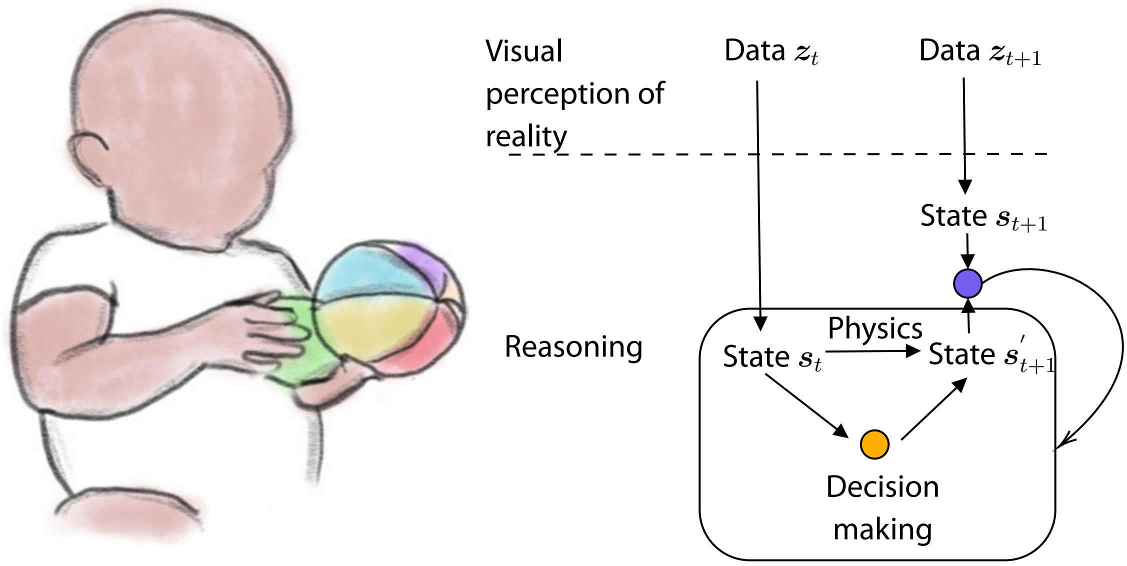


Figure 1.1: Learning cycle. Visual inputs are perceived and data is converted into information for reasoning. At this stage, learned physical models inferred from experience are used for the prediction of the future state and decision-making in critical situations. z_t represents the information available about the dynamics. In contrast, s_t represents the full state of the dynamics at a specific time step

bias the set of assumptions that we impose in the learning process so that the solutions fulfill those conditions.

As a breakthrough for fluid dynamics understanding, recent studies show how humans' intuition about liquids exhibits affinity to probabilistic fluid models [Bates et al., 2015]. This statement not only supports the claimed complexity of fluid reasoning and understanding but also that an approximation of this process utilizing physical modeling is possible to reach the same or a higher degree of accuracy and performance than human reasoning.

In contrast with these methods, learned simulation is a technique that learns continuously from the environment to perform the simulation and prediction of physics. Digital twins include these types of systems to be connected to the physical entity to integrate in real-time their future states. As a result, we have access to information about the real twin for the development of diverse applications.

1.3 Model problem

The proposed method is to be applied to the understanding of sloshing dynamics. In our daily routines, we mainly interact with solids and liquids, and liquid handling is part of an extended series of tasks that we continuously perform in particular. Infants with emerging cognition are already able to understand the behavior of such substances and their interaction with solids.

Liquids entail striking difficulty to be modeled. As a result, liquid manipulation is an arduous challenge for autonomous machine systems that rely on the proposed physics intuition models. Hence, the selection of this phenomenon will test our approach against different complex conditions, in particular non-linearity, dissipation, and partial access to information.

Among the existing list of countless fluid events—turbulence and laminar flows, among others—the present work focuses mainly on one of them: the sloshing dynamics [Ibrahim, 2005]. The slosh refers to the movement of a liquid that is inside a specific container. Typically, this phenomenon happens in an undergoing movement. The liquid necessarily has a defined free surface. Far from being negligible, the interaction between the waves and the boundaries of the vessel is a notorious problem in physics analysis. Modeling efforts are focused on estimating the motion of the free surface of the liquid. Such oscillations trigger critical forces in the container's walls. The modeling of sloshing dynamics is of great concern in a multitude of scenarios and fields [Huerta and Liu, 1988], see Fig. 1.2. The free-surface effect resulting from sloshing dynamics in open waters, for instance, can cause destabilization or capsizing of ships. Tanks partially filled can experience strong oscillations hard to be controlled. Computer graphics is one of the areas seeking to reproduce complex behaviors, traditionally studied through the Navier Stokes equations, to provide convincing representations of different case studies that range from flow divergence to splatter [Bender and Koschier, 2015] [Müller et al., 2003]. In what concerns this dissertation, we will focus on the problems that arise in robotic manipulation. These systems are of utmost importance in tasks such as packaging, assistance, cooking, or industrial applications.

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. When the container is squared the problem is approximated to a 2D description. Although good results are obtained, this assumption is no longer valid for cylindrical vessels. In these cases, a 3D description is required for an accurate result.

The model is usually coupled with control algorithms that attempt to ensure a slosh-free solution. Hence, the simulation should accomplish near or strict real-time performance. The aforementioned applications in control theory usually rely on coarse approximations of the slosh. The level of description required is the one that models efficiently the motion of the free surface to apply appropriate actions in the manipulation.

Equivalent mechanistic models fulfill the prerequisites of accuracy in the results of the models combined with the simplicity of controlling [Guagliumi et al., 2021] [Moriello et al.,

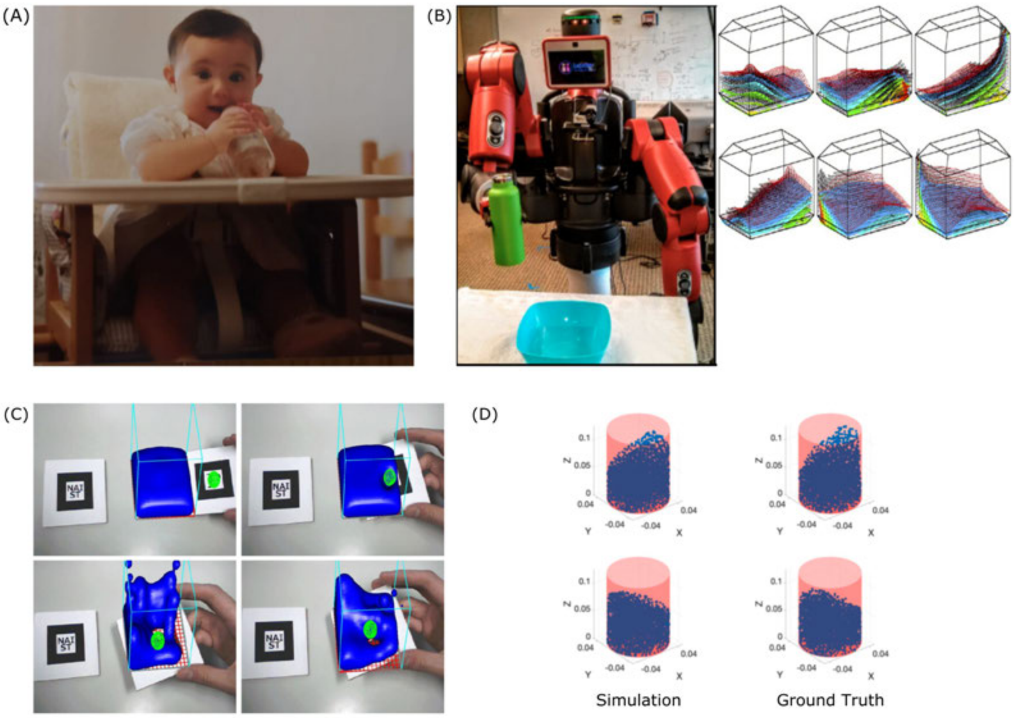


Figure 1.2: Examples of slosh. (A) Baby learning fluid dynamics through interaction. In manipulation, the slosh is one of the critical concerns to pursue a task. (B) Examples of study of the slosh in robotic manipulation to mimic human capacities [Schenck and Fox, 2018a] and analysis of wall forces in tanks undergoing sloshing [Kim et al., 2004]. (C) Visualization of sloshing dynamics through augmented reality to enrich the real environment [Fujisawa and Kato, 2009] (D) Approximation of the sloshing dynamics with the smooth particle hydrodynamics theory to model the perception system [Moya et al., 2019].

2018]. Although this technique is widely adopted in the community, a new paradigm emerges to combine the modeling and control of the slosh with artificial intelligence. The purpose is clear: while standard techniques are optimal, research can take a step forward in the development of new applications that include the independence of reasoning with smart use of data that can unveil new insights in the information for decision making. This is known as intelligent liquid handling [Kroemer et al., 2021] [Matl et al., 2019]. [Schenck and Fox, 2018a] employed convolutional networks to learn the physics of pouring liquids. Noteworthy, these predictions have no guarantee to comply with the laws of Navier-Stokes, for instance. This application is found especially relevant in the development of aiding systems for disabled or elderly people, or industrial applications.

1.4 Problem description

Physics perception involves three main paradigms. First of all, data needs to be acquired and processed. In human physics perception, we mainly work from visual inputs, i.e. images taken by our eyes. We structure and analyze them to find patterns and features from the information captured. Secondly, we need to build fast and accurate models that fit the complex events that we sense to learn real-world physics. Finally, models of physics and information are integrated for decision making and extrapolation of physical priors to develop our *common sense*. Adaptation and learning of new materials from observation, and using previous knowledge as a learning bias, is one of the applications of the latter statement.

The result of the thesis is a method for learned simulation for sloshing dynamics built upon the prerequisites of the agents involved in scene understanding to design a fully operative system. We consider a machine system that learns to perceive and make predictions of fluid dynamics with certain state variables that describe the state of the fluid under strict real-time constraints. The aforementioned system is connected to the real entity to acquire information and replicate real physics. Hence, the objective of the research is the development and assembly of a digital twin that performs learned simulation real-world physics to provide an interpretation of the scenario. It will enrich itself with the data acquired in the sense that it will enhance itself when outputs deviate from reality, improving its accuracy.

This is to be achieved by profiting from new advances in artificial intelligence, focusing on computer vision and machine learning. In particular, manifold learning and neural networks (NN) offer flexibility for the design of this application, which we enrich with knowledge of physics to obtain meaningful solutions.

1.4.1 Data acquisition

Perception and tracking of liquids are crucial elements for the success of the proposed statement. Nevertheless, its implementation with the simulation system is not trivial. Common liquids, as the examples presented in this thesis, entail different challenges. These are primarily rooted in the lack of texture (non-Lambertian properties) and translucent characteristics of some of them. This fact makes perception tasks especially difficult for commodity cameras, both monocular and stereo. Another factor that may affect fluid tracking is the container. Opaque glasses show liquid occlusions, and transparent vessels show distortions due to the diffraction properties of the material. Especially from certain angles, the method can be acutely compromised. Conversely, these threats are continuously addressed by the computer vision community, demonstrating the possibility to achieve the connection between the model proposed and the real entity.

In this thesis, we focus on the perception and tracking of the container and the free surface of the liquid of interest, which is performed with a stereo camera. These systems include two or more lenses to replicate the binocular nature of human vision. Although current monocular systems achieve high performance, stereo cameras are more convenient for fast and accurate 3D reconstruction of pixel coordinates. Many current cameras already incorporate functionalities for tracking and depth estimation. First, computer vision is employed to build an augmented reality application that correlates the movement of the glass that triggers the slosh of the liquid with the sloshing model proposed. Due to the lack of texture of the container, we add features to be detected to estimate the movement of the glass. Secondly, we track the free surface of the fluid to relate directly to the real entity with the model and execute correction and adaptation.

1.4.2 Real-time interaction

Another important component in the design of the perception system is the speed in both information interpretation and simulation. Liquids are deformable bodies. This fact entails added complexity to the problem. Although we employ a coarse-grained description of the fluid at a mesoscale, the full state of the volume is still remarkably high dimensional. Thus, it is described by a high number of degrees of freedom.

Consequently, time is a bottleneck for the communication between connected agents. Even in a low data regime, the perception loop could be jeopardized by the data required, the computational cost in training, and the time consumption during implementation due to the high dimensionality of the problem. The commodity camera that we employ to connect the algorithm with the real entity of the liquid performs at a frequency of $30-60$ Hz. Hence, the simulation loop must ensure real-time response and connection. A solution could be to simplify the employed description, but we compromise the accuracy and generalization of the test case. In addition, there has been considerable development in computer systems. However, we aim at developing a system that could be part of a portable gadget, for the ease of possible implementation. Another challenge that arises is that, considering the high dimensionality and complexity of the problem, convergence to infer correlations could be threatened. Data, especially that acquired from real environments, includes noisy measurements and, depending on the description used, it may be difficult for the algorithm to extract meaningful features and patterns from data.

As models of social, dynamical, or biological systems become more and more complex, model order reduction (MOR) is an appealing solution to prevent systems from being overwhelmed by the curse of dimensionality. What is more, they unveil useful instances of data to efficiently pursue different applications. Contrary to a rudimentary simplification of the dynamical description, model order reduction preserves the integrity of the insights of data to maintain the information in the reduction. A-posteriori MOR utilizes data to find a low-dimensional manifold where the dynamics are embedded. POD [Ly and Tran, 2001] is one of the most popular methods included in this category. Due to the ease of implementation and efficacy in many applications, it has been also a source of inspiration for the development of derived techniques. Conversely, non-linear model order reduction techniques such as kernel PCA ($k-$PCA) [Schölkopf et al., 1998], Locally Linear Embedding (LLE) [Roweis and Saul, 2000], Topological Data Analysis (TDA) [Wasserman, 2018] or Isomap [Tenenbaum et al., 2000], are suitable for problems of considerable complexity and strong nonlinearities.

There is a great variety of works to be explored in the field of fluid simulation facing new complexities and problems. Probably POD is one of the most employed methods in model order reduction due to its ease of application [Ortali et al., 2020]. Conversely, provided the highly non-linear nature of many complex phenomena, there is scope for the employment of techniques that exploit non-linear correlations to fit the characteristics of the system [Shvartsman and Kevrekidis, 1998] [Rowley and Dawson, 2017] [Chaturantabut and Sorensen, 2010] [Treuille et al., 2006] [Xiao et al., 2014] [Rowley, 2005], also POD adaptations [Kang et al., 2015] [Ahmed et al., 2021].

Such is the interest in this topic that deep learning has made, and continues to make, impactful contributions to this community. Autoencoders [Goodfellow et al., 2016] are a specific type of neural network that achieves compression of information on a latent manifold. They are presented combined with diverse types of structures. For instance, they have been explored in conjunction with feedforward nets [Hernández et al., 2021a] [Hesthaven and Ubbiali, 2018] [Chen et al., 2021a] [Taira et al., 2020] [Erichson et al., 2019] [Xie et al., 2020], convolutional neural networks (CNNs) [Hinton and Salakhutdinov, 2006] [Murata et al., 2020] [Kim et al., 2019] [Fukami et al., 2021] or graph architectures (GNNs) [Ranjan et al., 2018].

We explore different manifold learning techniques applied to sloshing to analyze their performance in data-driven modeling of complex physics. In addition, we study possible applications resulting from the insights that may come up in the low-dimensional manifold.

In spite of the effectiveness of MOR, we usually encounter observability issues that complicate data collection of the required state variables for the physically informed description. Self-supervised learning provides a framework to explore and develop more realistic and practical applications from not labeled, or partially labeled, data to learn to have a deeper understanding of data available to recover the unavailable information. [Callaham et al., 2019] [Sun and Wang, 2020] use strategically placed sensors that acquire data to recover the full set of quantities from sparse observations. Erichson et al. [Erichson et al., 2020] propose the use of shallow neural networks for reconstructing fluid flows. Lye et al. [Lye et al., 2020] estimate the unknown input parameters in turbulent flows from observables. In contrast, we propose un unveil the hidden features of the missing dynamics from the study of the history of the free surface.

1.4.3 Physics predictions with learned simulators

Data-driven modeling of physical systems is one of the major fields in current research. The era of the 4.0 industry and Big Data triggered this phenomenon that profits from acquired information to learn insights that conventional models cannot reach. This is known as inductive reasoning, where observations lead to broad conclusions. The development of simulation engines that resemble human perception has witnessed great advances in this line. However, to be useful, there is one main quality that they must fulfill: trustworthiness. There exist accurate purely data-driven models with a strong black box character. However, they lack generalization when they encounter new situations out of the bounds of the training dataset. In addition, we cannot ensure the physical soundness of the proposed solutions.

Many of the current works have been developed upon a black-box structure that shows limitations and lack generalization. These algorithms depend on the available data, whose quality and quantity will dictate the insights learned about physics. This situation occurs similarly with people; scarcity of experience may induce the misinterpretation of what a user senses through vision and hearing. But, as mentioned before, we store and enrich progressively the models that we have learned of the real world. The philosopher Immanuel Kant claimed in his book Critique of Pure Reason (1781) [Kant, 1908]: "We see things not as they are but as we are". Hence, we are biased when learning new concepts and ideas. By guiding the knowledge acquisition with the help of well-known physics inductive priors, we expect to give the method extra flexibility in terms of generalization, reducing the necessity of data, and capacity for adaptation to new scenarios.

The present work is intended to contribute to the field of physics perception by incorporating an added value to the process: our previous knowledge about physics, established in the form of commonly accepted physical laws. The major purpose of this addition is to develop a data-driven system able to replicate physics intuition with physically sound results.

Fig. 1.3 represents a scale of methods with regard to the knowledge and data used in the development of applications. Physics-informed machine learning is an intermediate point in this scale. It encompasses techniques that complement the lack of data with inductive biases [Battaglia et al., 2018]. This term refers to information about the problem imposed in the learning algorithm. It guides the optimization towards specific solutions that respect the imposed limitations, bounding the convergence to a specific structure. Consequently, algorithms reach more generalizable and smarter structures that ensure the physical consistency of the results in a wider spectrum of situations. A widely known example of this theory is Physics-Informed Neural Networks (PINNs) [Raissi et al., 2017]. From symbolic learning, [Brunton et al., 2016], to those that focus on Hamiltonian systems [Hesthaven et al., 2020] and thermodynamics [González et al., 2019b], this field in continuous growth includes a plethora of techniques that cover very different casuistries.

Despite the variety of techniques that this framework gathers, the method must show flexibility and ease of implementation in a simulation loop connected to real measurements. Thermodynamics adapt to the inherent dissipative nature of real phenomena and allow for modeling rather complex behaviors.

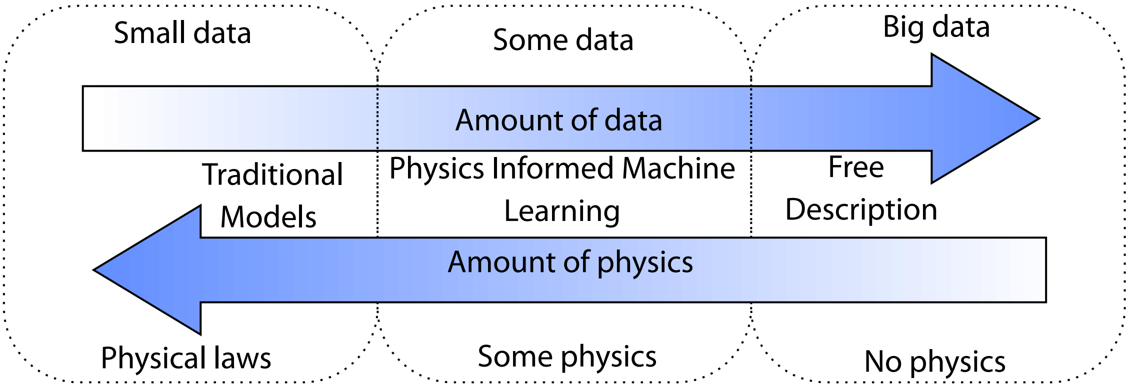


Figure 1.3: Data-driven learning scale, adapted from [Karniadakis et al., 2021]. Methods are conditioned by the amount of data available and the physics included in the learning algorithm. Traditional dynamical models (i.e. Navier Stokes) require a little data for fitting, while big data scenarios usually employ purely data-driven algorithms. Physics-informed data-driven methods find a balance between the data provided, especially when we work in small data regimes, combined with physics to guide learning.

The General Equation for the Non-Equilibrium Reversible-Irreversible Coupling, abbreviated as GENERIC, is a metriplectic formalism developed by [Grmela and Öttinger, 1997]. It expresses the time evolution of dynamical systems from energy and entropy potentials, which combined describe the problem's energetic state from a set of state variables. This formalism can be discretized and inferred from data to design a data-driven algorithm that fulfills the principles of thermodynamics by construction.

This formalism is the foundation of the algorithm proposed in this dissertation. It will be employed to learn sloshing models from computational data. In addition, it is to be coupled with data acquisition systems to the real liquid to integrate the fluid evolution in time and perform the correction of the model itself when encountering new, previously unseen, liquids.

1.4.4 Closed loop simulation and real-world interaction

A closed-loop simulation is defined as a *feedback* system. The algorithm retrieves output information to the model to maintain it stable and accurate. This theory is mainly applied to control problems, such as those involving reinforcement learning. Reinforcement learning (RL) proposes a series of techniques to perform control and model correction through interaction. Its use is widely known in self-driving cars or robotics, where it is applied to learning to adapt to previously unseen situations through the reward of desired behaviors.

In the case presented in this thesis, our physics perception problem can be formulated as a RL problem, where the correct predictions of the motion of the free surface are rewarded to correct the whole model and reach high prediction accuracy based on the observations captured with the camera.

1.4.5 Augmented reality representation

In contrast with artificial intelligence (AI), which focuses on the independence and capacities of a machine individually, intelligence augmentation (IA) enhances human capacities by providing extra knowledge to the user. The goal of perception systems is not only to interpret real events for their machine tasks but to show additional relevant information that the user may employ for decision making. Augmented reality's main purpose is to bridge the gap between the real asset and its digital twin. The model outputs data and, in a simplistic way of thinking, those are but mere numbers pending interpretation. Data must be presented in a user-friendly interface that facilitates assimilation by humans. The output represented in augmented reality not only consists of reconstructing the evolution of the dynamics but also of additional information not accessible at first sight that could be useful in critical scenarios.

This application is strongly aligned with real-time requirements. If the added information was not available at the time of the decision, the output of the simulation would be trivial in this operation.

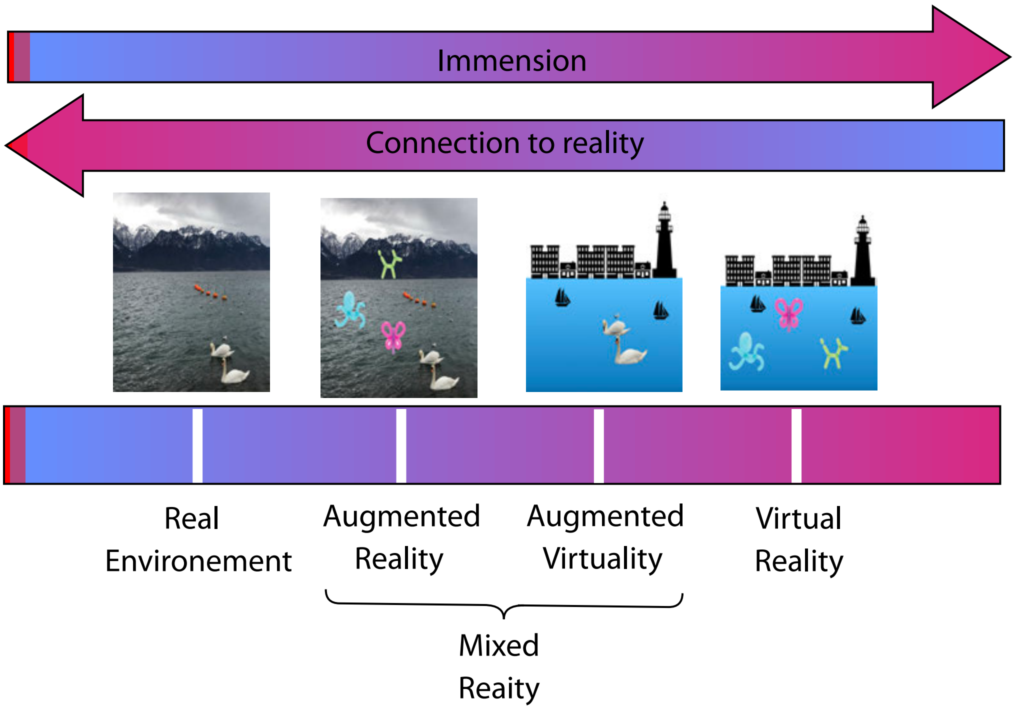


Figure 1.4: Reality-virtuality scale. It shows the progressive addition of virtual elements to the real setting. Augmented reality refers to the addition of virtual elements to reality, while augmented virtuality includes information from the real world in the virtual setting. Mixed reality operates in this range to perform the interaction between both realities.

Virtual immersion progressively introduces virtual elements into reality to mix both environments until the limit of total immersion in virtual reality displays, as seen in Fig. 1.4.

Augmented reality is the framework where a real-world environment is enhanced with virtual information, enabling the interaction of virtual objects in the reality [Moreland et al., 2013]. Augmented reality not only refers to the mix between reality and virtuality but also to the real-time behavior of objects and their correct positioning of them in the real 3D space.

AR tools have been successfully implemented for nonrigid representation, such as aerodynamics [Badías et al., 2019], deformable objects [Badías et al., 2018] [Badias et al., 2021] and contact problems [Badías et al., 2020]. [Fujisawa and Kato, 2009] propose an augmented reality interface for sloshing simulation. However, due to the complexity of this problem, they make use of GPU to be able to compute the solution and representation in real-time.

This work proposes the implementation of augmented reality for representation and interaction with the cognitive digital twin as a friendly interface for the user. In addition, the proposed system will output augmented information about the internal state of the fluid which can be used in other applications.

1.5 Data-driven computational mechanics

Computational mechanics has traditionally relied on mathematical approximations capable of accurately describing a plethora of physical phenomena. Just to cite a few, we could think of the widely used Hooke law, or Navier-Stokes and Bernoulli for the description of fluids. In contrast, data-driven computational mechanics propose the use of experimental data to perform the computational simulation of systems of interest, which might lead to a higher adaptivity to the processes described. In this area, machine learning propose several techniques to be employed in this task. Machine learning is the branch of artificial intelligence (AI) in which algorithms are trained with data to learn patterns to make predictions. Some of the most popular approaches in this field make use of Neural Networks (ANNs) [Oishi and Yagawa, 2017], but there are others based on bayesian learning [Welling and Teh, 2011] [Rezende et al., 2014] [Zhang et al., 2018] and Gaussian Processes (GPs) [Wang et al., 2005] [Wang et al., 2007] [Buisson-Fenet et al., 2020], random forest [Ladickỳ et al., 2015] [Jain et al., 2021], support vector machines [Ershadnia et al., 2020] [Drezet and Harrison, 1998], or nonlinear regression algorithms [Kanno, 2018].

The aforementioned algorithms play a capital role not only in the prediction of future states in the simulations, but in the design, optimization, and control of systems. Therefore, these approximations need to solve some issues to be sufficiently meaningful. Linear systems are easily characterized, but even small nonlinearities are challenging issues in data-driven computational mechanics. Other problems encountered are the curse of dimensionality, where high dimensional descriptions make it difficult to converge and learn, noise, or multiscale dynamics, among others.

The primary objective of supervised learning algorithms is to acquire the generalization ability to push the boundaries of a model to tackle previously unseen situations. Only under these conditions, we could approximate a problem with a sufficiently general trustworthy solution. Nevertheless, we still cannot ensure the physical validity of the results. If machine learning is tailored so that the results are physically consistent and interpretable, the model inferred from data can give new insights about the system characterized for its understanding.

This is a recent but prolific field of research, where studies try to reach higher generalization and interpretability of the solutions obtained to mirror classical physics. Efforts have evolved towards the use of a new branch of methods to include knowledge coming from the structure and/or the physics of the problems to guide the regression towards learning meaningful structures. These restrictions are denominated inductive biases and we distinguish two types of them [Karniadakis et al., 2021].

- *Observational biases*: As we have mentioned before, traditional black-box algorithms depend on the quality and the amount of data to learn sufficiently general laws. *Observational biases* rely on the physical meaning of data to, with enough snapshots, learn general approximations [Lu et al., 2021a] [Kashefi et al., 2021] [Yang and Perdikaris, 2019]. However, in optimization problems, there is no unique solution that fits the provided data.

- *Inductive biases*: In cognition, we base our learning on some acquired priors. For instance, any child knows that any physical system is subjected to gravity, even if they do not know what it is or its value. In the same way, we could impose known priors of the systems into the learning process to infer correlations that adapt to the problems. These algorithms are based on known learning schemes (ANNs, GPs, etc...), and tailored to include the information that we know a priori. This is called *inductive bias*.

Inductive biases group two subclassifications, as displayed in Fig. 1.5. *Relational inductive biases* are those that take into account the properties of the structure of data [Battaglia et al., 2018]. For instance, symmetries are those transformations that do not affect the properties of the system. If we know the characteristic symmetries of the given dataset, they should be considered in the optimization to guarantee the compliance of the results. For instance, in image learning, we must respect the translation symmetry, as it is done in convolutional neural networks. This field is explored in geometrical deep learning, where the authors suggest a broad mathematical framework in the domain of deep learning and ANNs to derive algorithms that adapt to geometries that go beyond euclidean domains, such as graphs, manifolds, or point clouds, to name a few [Bronstein et al., 2021].

There are also inductive biases, called learning biases, that consist in imposing physical priors to fulfill some theories, structures, or basic principles learned along centuries of research and study of mechanics and dynamics [Karniadakis et al., 2021]. These two types

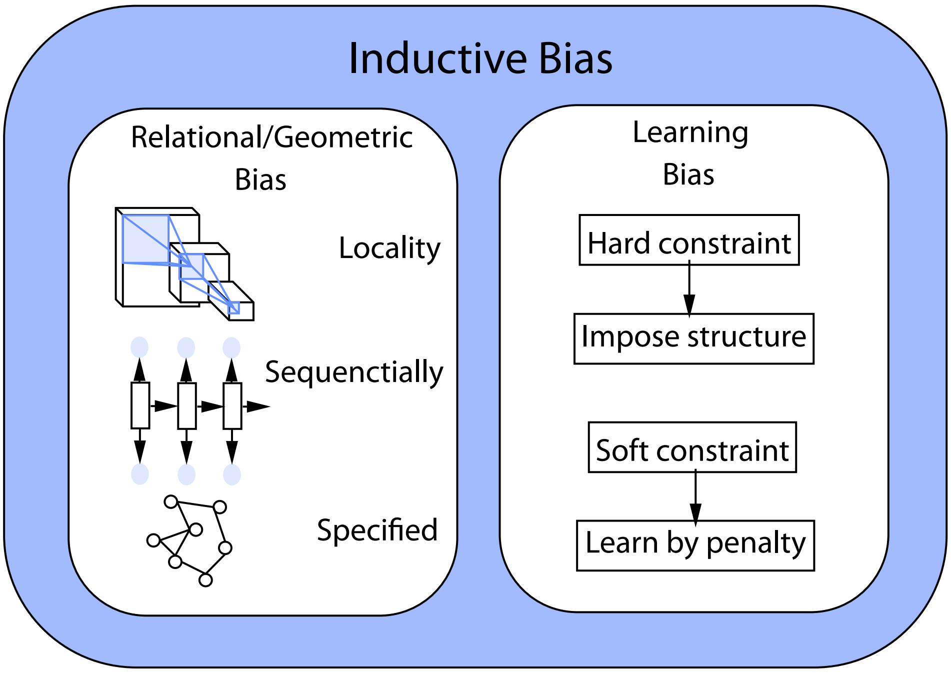


Figure 1.5: Inductive bias classification for the different types of knowledge that can be included into learning described in [Karniadakis et al., 2021].

of priors are not exclusive. They can be combined, but it is a field that has started to be explored [Hernández et al., 2022] [Bogatskiy et al., 2022]. In this work, we focus on the application of learning biases, particularly on those based on thermodynamic principles to learn meaningful and physically sound schemes from data.

Biases can be imposed as hard or soft constraints. Physical laws can be imposed as hard constraints, either in the neural network architecture, which is still an open problem, or during the regression, which may lead to a difficult convergence. On the other hand, soft constraints are more flexible in exchange for tolerance in learning the approximation.

The benefits of including *learning biases* are quite appealing. In scenarios where data collection is difficult, it might be challenging to learn general rules. We complement the lack of data with information to manipulate it smartly in the learning algorithms. This is called the smart data paradigm [Chinesta et al., 2020], where instead of focusing on gathering data, we emphasize how it is used to reach smart structures, which are specific reached solutions of the optimization that exploit known information about the problem.

1.5.1 Classical Regression

There has been a long evolution in physically sound data-driven learning over the last years. [Ibáñez et al., 2018a], followed by other works in the field [Ibáñez et al., 2017] [Ibáñez et al., 2019] [Ibáñez et al., 2020], abandon the idea of using constitutive equations to propose employing the constitutive manifold where the dynamics are embedded based on a-posteriori MOR. Learned from data, this manifold has locally linear submanifolds to approximate the predictions for new input values, such as stress and strains [Ibáñez et al., 2018a].

Some other incipient works were dedicated to imposing conservation laws into the learning optimization to select the solution that best fitted those conditions in material descriptions [Kirchdoerfer and Ortiz, 2016] [Kirchdoerfer and Ortiz, 2018] [Stainier et al., 2019] and fracture problems [Carrara et al., 2020] [Carrara et al., 2021].

The Koopman Operator and the related Dynamic Mode Decomposition (DMD) [Kaiser et al., 2018] [Kutz et al., 2016] [Proctor et al., 2016] also arose interest in the research community for data-driven learning. In contrast with POD-like methods, which compute spatial modes, the DMD mechanism is based on the computation of spacetime modes from data, which are eigenmodes and eigenvalues of a linear approximation to the nonlinear governing operator of the dynamics [Alla and Kutz, 2017] [Proctor et al., 2018].

Symbolic machine learning focuses on learning symbolic expressions for dynamical systems that fit their behavior [Zames et al., 1981] [Lui and Wolf, 2019]. Eureqa [Bongard and Lipson, 2007] propose a framework to infer a posteriori nonlinear symbolic approximations of dynamical systems. They first perform *partitioning*, which consists in optimizing the equation to describe each variable separately to, in a second step, simplify the structure of the model. SINDy (Sparse Identification of Non-Linear Dynamics [Brunton et al., 2016] is based on the hypothesis that the dynamics evolve in a sparse, low dimensional, function space. In the optimization phase, the algorithm penalizes the model complexity to find a combination of a few functions that are the coefficients of the solution governing equation. Similarly, [Flaschel et al., 2021] search for interpretable constitutive laws for hyperelastic materials by imposing compliance with conservative laws.

The method *sparse Proper Generalized Decomposition* ($s-$PGD) [Ibáñez et al., 2018b] [Sancarlos et al., 2021c] could remind also of this kind of methods. It begins with the same ideas from a PGD approximation of the objective function into a sum of one-dimensional functions. In this case, the approximation of the separated form is inferred from data to minimize the distance to the objective function. This method has also been employed for inferring the correction term in hybrid twins modeling [Moya et al., 2020a]. Sancarlos et al. [Sancarlos et al., 2021a] [Sancarlos et al., 2021b] propose variations of this technique: the regularized sparse PGD (rs-PGD) and the doubly sparse PGD (s^2-PGD).

1.5.2 Neural Network approaches

Probably the most popular techniques in the framework of physics-informed machine learning are those built upon neural networks. The term *physics-informed neural networks* (PINNs) was first coined by [Raissi et al., 2017], and it is probably one of the most known representatives in the field. The empirical risk is the evaluation of the performance of the algorithm on training data. This evaluation guides the training of the algorithm. In this case, the empirical risk of the optimization includes the evaluation of the discrepancy between the predictions and ground truth and the penalty if the PDE constraints are not fulfilled as a residual in the loss as a soft constraint. They have shown great effectiveness in different fields and applications [Chen et al., 2021a]. For instance, [Raissi et al., 2018] encode partial differential equations for problems such as the heat equation, demonstrating long-time stability in the integration.

PINNs have inspired several works in neural networks [Morton et al., 2018] [Ayensa-Jiménez et al., 2021], bayesian learning [Dandekar et al., 2020], and Gaussian processes [Raissi and Karniadakis, 2018] [Chen et al., 2021b] [Pang et al., 2019] with parameterized governing equations as a constraint for physically sound schemes. [Lu et al., 2021b] impose PINNs as a hard constraint in learning. Ayensa et al. [Ayensa-Jiménez et al., 2021] [Ayensa-Jiménez et al., 2020] implement not only physical laws as hard constraints in the neurons, but also internal states as soft constraints, into learning.

In the field of symbolic learning, [Cranmer et al., 2020b] propose a learning algorithm that seeks analytic relations based on exiting analytic equations of physics to build new expressions that are more interpretable.

These approaches also provide stability and interpretability of the solution, a convenient characteristic not only in interpolation but also in extrapolation out of the database range. [Miller et al., 2020] perform an analysis of the forecasting efficiency of physically informed machine learning in the particular case of Hamiltonian problems considering different approaches. By evaluating the predicted energy and state variables error, they support the outperformance of physics-informed methods over conventional black-box neural networks.

1.5.3 Machine Learning for Fluid Dynamics

Complex flow simulation, especially under strict real-time specifications, remains an open research field. The main challenge lies in the balance between accuracy in the reconstruction and the physical meaning of the results, and the computation time of the time integration. Usually, the computational resources required for complex descriptions exceed real-time constraints, and the extreme simplification of the description would cause the loss of relevant information and physical consistency.

Data-driven descriptions are appealing approximations to fit these behaviors, especially with real measurements, due to their high dimensionality and complexity [Morton et al., 2018] [Wiewel et al., 2019] [Tompson et al., 2017] [Schenck and Fox, 2018a] [Ummenhofer et al., 2019] [Sanchez-Gonzalez et al., 2020]. [Ladickỳ et al., 2015] proposed a GPU implementation of fluid simulation consisting of a forest regressor [Cutler et al., 2012] with an adaptive time integration step, which eliminated the time step restrictions for complex simulations. Closely related, Byungsoo [Kim et al., 2019] explored reduced representations of smoke and different types of liquids with CNNs from velocity fields. Yunzhu Li et al. [Li et al., 2018] study the (DPI-Nets), ANNs for fluid simulation from hierarchical particle interactions in smooth hydrodynamics discretizations. Although these methods proposed visually admissible results, solutions lack physical meaning.

[Brunton et al., 2020] make a review of these recent advances in machine learning applied to fluid mechanics and dynamics. It stands out the need for these systems to overcome challenges such as generalization and interpretation. Particularly, there is a growing interest in machine learning strategies that satisfy well-known principles of physics and do not construct black-box models whose application to previously unseen data does not guarantee to comply with basic principles such as energy conservation or equilibrium.

Physically sound data-driven modeling has also reached the field of fluid computational modeling. Fluid learning is a rather prolific field due to the complexity that it entails and its wide presence in many phenomena of utmost importance. [Alla and Kutz, 2017] [Proctor et al., 2018] show an application of DMD for fluid dynamics learning from velocity fields. [Kim et al., 2019] propose a generative network that learns the velocity field of fluids (smoke and liquids). They work in parallel with a second network.

ANNs are an extended tool for its approximation [Kim et al., 2019]. [Tompson et al., 2017] [Miyanawala and Jaiman, 2017] apply CNNs to characterize 2D and 3D fluid dynamics. In the case of [Bukka et al., 2021], the authors distill the dynamics of unsteady flows with RNNs. Following the same spirit of the latter work, [Wiewel et al., 2019] employs specifically LSTMs networks in reduced order manifolds, while graph neural networks are becoming popular in this field [Sanchez-Gonzalez et al., 2020]. Regarding physically informed deep learning, there are also works related to the study of fluids. [Mao et al., 2020] apply PINNs to high-speed flows. The work of [Gao et al., 2021] presents an approach for learning PDEs from Physics-informed CNNs.

1.6 Objectives

The main global objective of this work is to advance in the development of tools to use simulation as the engine that performs physics perception. The present dissertation breaks down the three learning steps of perception to design a new augmented intelligence algorithm, able to perceive, simulate, and reason.

We have chosen the sloshing dynamics as a model problem, but other problems of interest could have been chosen as well. The global objective is achieved by the consecution of the following objectives:

- Designing a learned simulator from full-field data (full description of the dynamics) of sloshing applicable to different liquid behaviors under strict real-time constraints.
- Guaranteeing, by construction, basic physical principles to ensure the obtention of physically sound results, stability, and generalization of the simulation.

- Implementation of the digital twin with information coming from the real item and perform reality-virtuality interaction.
- Integration of simulation algorithms and visual techniques for the development of augmented reality and augmented intelligence applications.
- Overcoming the limitations of partial measurements to ensure the physical consistency of the approach when learning from limited data.
- Developing a reinforcement learning correction system to achieve adaptation to new liquids and changes in the environment through reinforcement learning.

1.7 Thesis structure

This dissertation is structured in 6 chapters that walk through the process of concept design, training, and application of the proposed framework, progressively addressing new challenges in the development of the perception system. At each development step, shown in Fig. 1.6, we include an additional asset to enrich the learned simulation, driving towards a more complex, but complete, perception system.

- Chapter 1 exposes the motivation and objectives of this dissertation. We put in context the addressed perception problem, and how different disciplines interplay to build a physics perception algorithm.
- In Chapter 2 we perform an analysis of the application of manifold and machine learning techniques to the sloshing dynamics under study. We propose different linear and non-linear techniques to compare their performance in the described problem. We evaluate the results in the reduction capacity to embed the non-linearities distilled from data and the stability during the integration of the dynamics in the low dimensional manifold. The output of this chapter is the foundation of the learned simulator.
- Based on the results from Chapter 2, Chapter 3 describes the implementation of the learned simulator in a digital twin of a real glass in real-time. We perform visual interpretation for liquid understanding and perception of different types of liquids.
- In Chapter 4 we address the limitation of information from a commodity camera's experimental measurements. In this step, we record the real liquid, from which we only have access to the free surface. We extend the modeling to deep learning architectures that achieve high accuracy and offer feasible alternatives to undertake partial information.
- Chapter 5 tackles the final step in a close loop algorithm: the comparison between the simulation output and the ground truth. We extend the problem from a digital

twin to a hybrid twin to include the self-correction needed to adapt to previously unseen liquids.

- Chapter 6 shows an overview of the results obtained and conclusions extracted from them. In addition, we present the publications of the content of the dissertation.

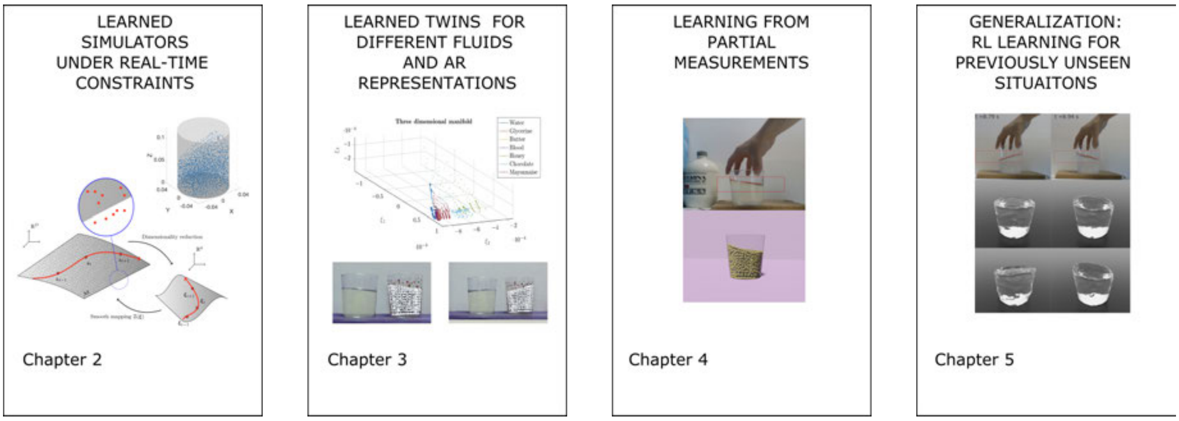

LEARNED SIMULATORS UNDER REAL-TIME CONSTRAINTS
Chapter 2
LEARNED TWINS FOR DIFFERENT FLUIDS AND AR REPRESENTATIONS
Chapter 3
LEARNING FROM PARTIAL MEASUREMENTS
Chapter 4
GENERALIZATION: RL LEARNING FOR PREVIOUSLY UNSEEN SITUAITONS
Chapter 5

Figure 1.6: Thesis outline. The development of the perception algorithm is broken down into four steps. A chapter is dedicated to each of them.

Chapter 2

A first attempt based on manifold learning

This chapter proposes several manifold learning techniques to describe sloshing problems based on data. The model is to be developed in a reduced order space under rigorous thermodynamics settings. We exploit several model order reduction (MOR) techniques. We employ Proper Orthogonal Decomposition (POD), followed by non-linear strategies (Locally Linear Embedding and Topological Data Analysis). All three distinct possibilities rely on a numeral integration scheme to advance dynamics in time. We show how the resulting method employs a few degrees of freedom, while it allows a realistic reconstruction of the fluid dynamics of sloshing under severe real-time constraints.

This work has been published in the following contribution:

- Moya, B., González, D., Alfaro, I., Chinesta, F., & Cueto, E. (2019). Learning slosh dynamics by means of data. Computational Mechanics, 64(2), 511-523.

2.1 Introduction

The inception of perception systems starts with the development of the learning algorithm that leads to the learned simulation. In other words, we first propose a fast and accurate algorithm for the simulation of the dynamics in real-time. We have established a data-driven framework for the design of the algorithm for this purpose. It includes two main contributions: the proposed algorithm must be physically rigorous while accurate and flexible to learn different types of behaviors, and the integration in time of the dynamics must respect real-time (30 Hz) response rate to enable the connection with real-world dynamics.

In this dissertation, we suggest a method that learns to replicate physics from observation, not only in the training phase but also in performing continuous interaction leading to learned simulation in a digital twin. We first develop a learned simulator based on data-driven techniques. We avoid black-box algorithms, which are unpredictable and lack generalization. We aim to impose physical knowledge, in this case the restrictions imposed by the compliance of the laws of thermodynamics, to ensure the physical consistency of the results.

Data-driven methods have drawn attention in recent developments. These, combined with machine learning assets, show high accuracy in replicating physics with data. However, the method proposed in this dissertation opens the black box of traditional data-driven learning for combining observations with well-known epistemic requirements.

In this attempt, we build a simulation engine that mimics de sloshing dynamics that will be the core of the future cognitive digital twin. Based on our previously stated hypothesis, we propose the use of manifold learning to infer the knowledge of the dynamics with purely data-driven methods in a low-dimensional manifold where the dynamics are embedded. We construct the slow manifold of the sloshing guided by the GENERIC formalism, which will ensure compliance with the laws of thermodynamics in the description of the dynamics evolution. [González et al., 2019b] introduced its implementation in data-driven modeling as an inductive bias, and it has been successfully applied in several fields [González et al., 2021] [Ghnatios et al., 2019].

The present work resembles the strategy developed by [Millán and Arroyo, 2013]. In that case, emphasis was put on the non-linear dimensionality reduction aspects of the technique. As will be done here, Millán and Arroyo developed an integration scheme in generalized (reduced) coordinates. A fundamental difference of the present work is that the integration scheme will be learned from data and, contrarily to the work of Millán and Arroyo, is valid for Hamiltonian as well as for non-conservative or out of equilibrium mechanics.

As a first try, we model the sloshing of water with data obtained under different initial conditions. The liquid is contained in an ordinary glass. The learning process could be extended to other container typologies, but in this case, we initially bound the problem to one geometry and one material. The geometry of the container is defined with diameter $b = 10$ cm, and it is to be subjected to sloshing forces. The geometry is displayed in Fig. 2.1. We consider a fluid volume that adapts to the shape of the glass, filled up to 7 cm.

The results obtained at this step will set the foundation for the development of models for a wider variety of liquids with diverse properties and the implementation of the digital twin from the analysis of the performance of the algorithm and the verification of the real-time constraint.

2.2 The dynamical systems equivalence

The problem of learning physics from data is formulated as follows [Weinan, 2017] [Ma et al., 2020]. Let $\{\boldsymbol{z}_i\}$ be input values sampled on a domain $S \in \mathbb{R}^D$, with D the dimensionality of each sample. We have n values in the domain in a time interval, and suppose that z_i is a time discretization of a dynamical system, where values are probably noisy. We also consider a regular time discretization for which $t_{n+1} - t_n = \Delta t$ for all n.

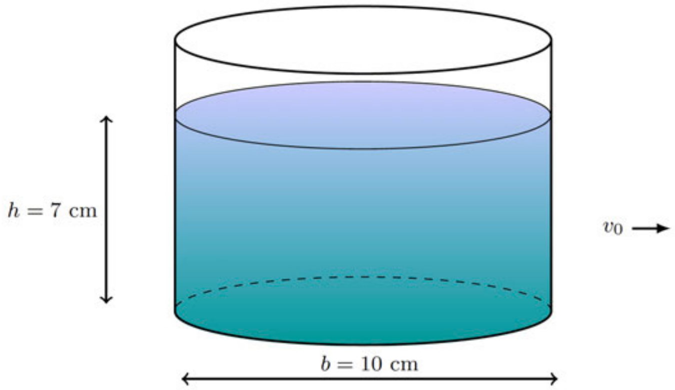


Figure 2.1: Geometry of the liquid and the glass. The problem is defined for one geometry and one type of liquid.

The learning problem presented in this dissertation is equivalent to a dynamic problem whose evolution is to be learned, where the correlation between a set of state variables and their rate of change is expressed as

$$\dot{\boldsymbol{z}}_t = \frac{d\boldsymbol{z}(t)}{dt} = f(\boldsymbol{z}(t)),$$

with

$$\boldsymbol{z}(0) = \boldsymbol{z}_0,$$

where z_t stands for a set of variables that describe the aforementioned system and its growth, and f, the mapping governing the evolution, whose form is to be learned from data.

The set of *state variables* defines the mathematical state of a dynamical system and includes enough information to predict its future behavior—not considering external inputs such as external forces. Some typical state variables are the position of geometrical points or velocity fields.

In dissipative problems and thermodynamical formulations, we require more information to characterize this behavior. Thus, additional state variables that capture this information are required, for instance, the internal energy. The set of selected state variables and their combinations, which result in different states, form the state space of the system in which we carry out the learning. The input-output correlation will be computationally approximated with a dataset of available measurements.

The set z_t are actually the coordinates of the smooth manifold of the dynamics for which the differentiable maps have been defined. f is learned so that the flow map $\boldsymbol{z}_0 \to z(T, \boldsymbol{z}_0)$ replicates the response, typically nonlinear, required to adapt to data and fit the dynamics. Since this is a machine learning problem where we know the input as well as the output of the approximation, it can be considered supervised learning.

As mentioned before, $\boldsymbol{z}_i$ obeys an unknown equation that defines its rate of change. The goal is to be able to predict the state of the system given $\boldsymbol{z}_n$ at time t_{n+1} by learning an approximation of the equation f. This function is approximated with pairs of inputs and outputs $(\boldsymbol{z}_n, \boldsymbol{z}_{n+1})$ to accommodate the behavior of the domain by deriving a surrogate model of the form

$$\boldsymbol{z}_{n+1} = u(\boldsymbol{z}_n).$$

Tuning the appropriate f thus becomes an optimization problem. We denote as u the proposed continuous function solution for the domain. Empirical risk minimization is the principle used in statistical learning to perform the optimization. We define the *empirical risk* R as a description of how well the algorithm is performing on training data to bound the optimization to improve the performance. The empirical risk is defined as a

$$R(f) = \sum_{i=1}^{N} \mathcal{L}(\boldsymbol{z}_{n+1}, f(\boldsymbol{z}_n)),$$

where $\mathcal{L}(\cdot,\cdot)$ is the metric that evaluates how well the function learned fits the dynamics, in the form of L_2-norm or more sophisticated evaluations. As a result, we minimize the error between the proposed solution and the ground truth

$$u = \text{arg}\min_{f} R(f).$$

The function f can be found based on different hypotheses with regard to the dynamical system to build different approximations of reality. The goal of data-driven modeling is to learn f in such a way that the flow map reproduces a non-linear function that fits the provided data by regression. In this context, there are several data-driven methods to be considered.

2.3 Data collection

We collect data of the behavior of water from the discretization of Navier-Stokes equations governing the fluid motion by applying the Smooth Particle Hydrodynamics method [Monaghan, 1992] as described in Fig. 2.2. Developed by Gingold and Monaghan, they propose a discretization into particles to approximate the dynamics of continuum media, considering solids as well as fluids. Thus, it provides a Lagrangian meshless description of the fluid. Particle discretization has been largely applied to analysis since particle-based methods offer good approximations to deal with complex fluids due to their efficient adaptability to complicated geometries. It is generalizable to the simulation of different types of fluids, and it provides a sufficiently good description of the dynamics balanced with the computational cost. Its application is demonstrated in a plethora of works in computer graphics.

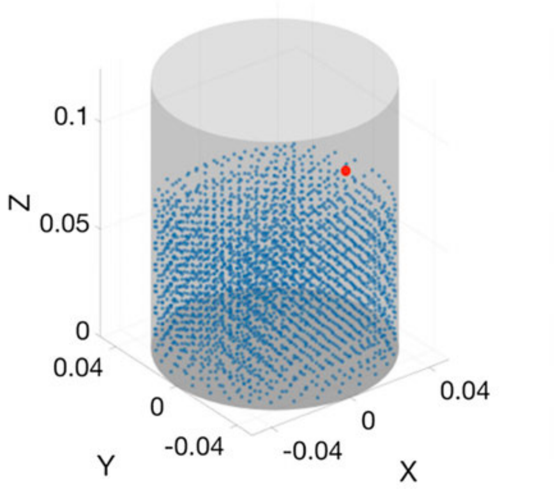


Figure 2.2: Cloud of nodes of the discretization of the liquid at rest, which is its initial state for the simulation of all velocities. There are 2898 particles in the discretization. Particle with label 1, for which some results are analyzed hereafter, is highlighted in red.

The fluid of our case of study is discretized into $M = 2898$ particles. It is well-known that for a Newtonian fluid the adequate set of state variables that characterizes a fluid particle is composed by its position $\boldsymbol{q}_j$, velocity $\boldsymbol{v}_j$, and internal energy E_j [Español, 2004], so that the set $\mathcal{S}$ of state variables is :

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

where $\boldsymbol{z}$ is 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 [Español, 2004] [Romero, 2009a] [Romero, 2010] [Romero, 2013]. An appropriate selection is required to ensure the thermodynamic definition of the system. In other words, the energy $E(\boldsymbol{z})$ can be written as a function of the selected variables.

The fluid selected for the development of the first test case is water, due to its generality, versatility, and wide presence in daily tasks. We describe it by using its density $\rho = 1000$ kg/m^3 and the $U_s - U_p$ Hugoniot form of the Mie-Grüneisen equation of state [Herrmann, 1969] [Carroll and Holt, 1972] in the commercial software Abaqus CAE (Dassault Systèmes).

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 of the dynamical evolution of the slosh is obtained for a time increment of $t = 0.00016$ seconds. 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. Depending on the reduction method imposed, more or fewer snapshots will be required.

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, $D = 2898 \times 7 = 20286$, justifying the need for a re-parametrization to embed the dynamics onto a lower-dimensional manifold. The selection of the new parametrization will be further discussed in the next sections, where different methods will be proposed.

The data matrix available for training is composed of the set of n snapshots selected grouped by columns,

$$\begin{bmatrix} | & | & | & | \\ \boldsymbol{z}_1 & \boldsymbol{z}_2 & \cdots & \boldsymbol{z}_n \\ | & | & | & | \end{bmatrix} = \boldsymbol{Z} \in \mathbb{R}^{D \times n},$$

that describes the time history of the sloshing movement.

2.4 Manifold learning techniques

We have performed some numerical simulations so that the dynamical evolution of the sloshing dynamics is described discretely. We have obtained n snapshots for the whole set of particles N representing the time evolution of each simulation for all the computed time steps. Manifold learning techniques look for the most distinctive structure in data unveiled from the provided data. Given some measurements, these methods study the hidden correlations among the state variables of each snapshot to ultimately find a representation of the same information but in a space of much lower dimension $d \ll D$. This hypothesis is represented in Fig. 2.3.

POD is a popular and successful technique in model order reduction. Conversely, the problem under study is highly nonlinear, and this fact encourages the exploration of other techniques capable of unveiling and capturing the non-linear features that can be distilled from data.

2.4.1 Proper Orthogonal Decomposition (POD)

POD [Ly and Tran, 2001] is widely known for its application to model order reduction for dynamical systems, and more precisely those involving complex fluid dynamics. Essentially, it finds an orthogonal projection to the tangent plane at a given point from the eigenvalue and eigenvector decomposition of the covariance matrix of the data provided. We hypothesize that the gathered information of the sloshing dynamics underlies hidden structures that can be exploited.

In this specific case, the dimensionality of the fluid description is greater than the number of snapshots collected. If ordinary POD was performed, the result would be distorted, showing highlighted values that are related to the size of the time dimension. In addition,

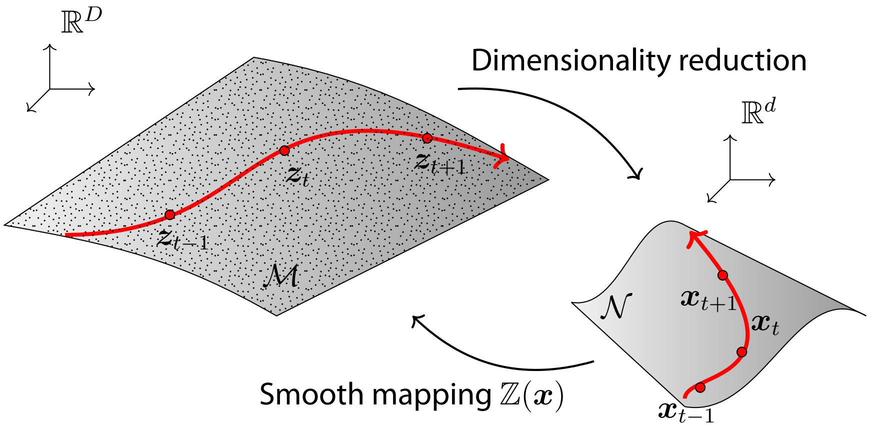


Figure 2.3: Hypothesis about the existence of a slow manifold $\mathcal{M}$ on which the fluid lives. The small dots represent the experimental data (snapshots) in a high-dimensional space $\mathbb{R}^D$. An arbitrary trajectory of the system in the phase space is represented in red. Dimensionality reduction is applied to project the data to an embedding space in $\mathbb{R}^d$, with $d \ll D$ where the system dynamics will be integrated.

the computation of the covariance matrix $\boldsymbol{Q} = \boldsymbol{Z}\boldsymbol{Z}^T \in \mathbb{R}^{D\times D}$ and the eigenvalues would probably be prohibitive in terms of time and space resources. We avoid this through the application of the method of snapshots. In this case, the covariance matrix is computed as:

$$\boldsymbol{Q} = \boldsymbol{Z}^T\boldsymbol{Z} \in \mathbb{R}^{n\times n},$$

to perform the eigenvalue decomposition

$$\boldsymbol{Q}\boldsymbol{v}_j = \lambda_j \boldsymbol{v}_j, j = 1, ..., n,$$

which would be equivalent to studying the right-singular vectors in Singular Value Decomposition.

The corresponding POD modes can be then computed. We perform the transformation from the computed orthogonal basis in the method of snapshots to the POD projection by a linear mapping:

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

The dimensionality of the low dimensional manifold is chosen from the analysis of the computed singular values. Top eigenvalues encode the main features detected in the dataset. On the contrary, the lowest eigenvalues correspond to trivial structures associated with noise and perturbations. We choose the highest-valued eigenvectors to build the new projection. They are chosen depending on the eigenvalue evolution. The number of modes is the minimum possible to preserve the most relevant features of the dynamics.

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 several modes higher than that value may be needed to ensure that we have an accurate integration scheme. This set of d vectors form the projection matrix $\boldsymbol{D}$, which works bi-directionally to offer a linear mapping between the physical space $\boldsymbol{Z}$ and the reduced basis,

$$\boldsymbol{Z} = \boldsymbol{D}\boldsymbol{X}.$$

The problem with a POD-based approach is that it provides 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.

2.4.2 Locally Linear Embedding (LLE)

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 [Roweis and Saul, 2000], considers that each point of the cloud will be interpolated by its K nearest neighbors with their pairwise euclidean distance. Local linearity will be assumed to exist in the neighborhood of each point. 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 of selecting 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.

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

$$\varepsilon(\boldsymbol{z}) = \sum_{i=1}^{n} \left| \boldsymbol{z}_i - \sum_{j=1}^{K} W_{ij}\boldsymbol{z}_j \right|^2 ,$$

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 add up 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 assumed 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 is the new coordinates $\boldsymbol{X} = \{\boldsymbol{x}_i\}, i = 1, ..., M$ of the points in the embedded space, which is calculated by the cost function related to the embedded space,

$$\Phi(x) = \sum_{i=1}^{n} \left| \boldsymbol{x}_i - \sum_{j=1}^{K} W_{ij} \boldsymbol{x}_j \right|^2 .$$

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, it could be reconstructed in the full space.

2.4.3 Topological Data Analysis (TDA)

Data science has recently been attracted by the sought of topological underlying structures in data to analyze its complexity. One of the milestones achieved by these techniques is their use for manifold learning purposes. Data has a shape, which can be interpreted as the slow manifold where it is embedded. One of the 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 to find the smooth manifold of the object of study.

TDA makes intensive use of persistent homology, which consists in the sought of topological features of data [Munch, 2017] [Wasserman, 2018]. 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 the 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 to a 1-simplex, or edge. To define a simplex of higher dimensionality, the M-points that would conform to it must be closer than R to every other point. The set 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

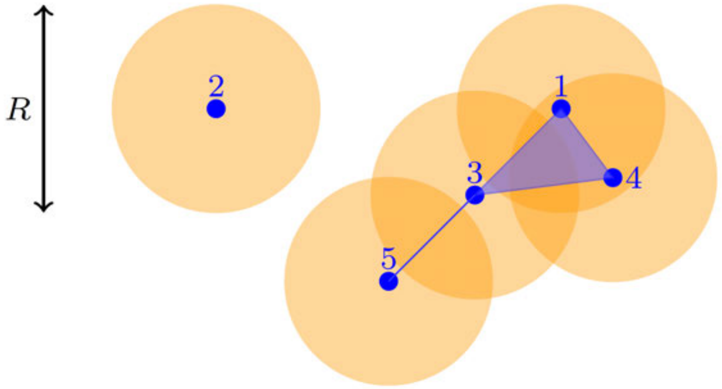


Figure 2.4: A sketch of the concept of persistence homology. By making R grow from 0, simplices (edges, triangles, tetrahedra) appear (respectively, when two, three, or four circles/spheres intersect) or disappear (when a hole collapses). Persistence diagrams collect the values of R for which these simplices appear or disappear.

disappears over 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. 2.4. For the represented value of R, node 2 is isolated from the rest of the 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 with noise in the data.

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 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) an interpolation scheme within the just obtained manifold structure of the data. Here, we will employ Kriging [Williams, 1998]. 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 $N(i)$. We calculate the embedded coordinates x_i of each point by minimizing the functional

$$\Phi(\boldsymbol{X}) = \sum_{i=1}^{n} \left| x_i - \sum_{j=1}^{N(i)} W_{ij}^* x_j \right|^2,$$

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

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 the necessary smooth mapping between the embedded coordinates and the physical ones:

$$Z : \mathcal{A} \subset \mathbb{R}^d \to \mathbb{R}^D,$$

$$\boldsymbol{x} \mapsto \sum_{j=1}^{N(i)} W_{ij}^*(\boldsymbol{x}) \boldsymbol{z}_j,$$

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

2.5 Dynamical systems with known properties

Works in machine learning for computational mechanics are not only restricted to algorithms based on PDEs constraints [Yang et al., 2018] [Hanuka et al., 2021] [Rath et al., 2022] [Yang et al., 2021]. [Schmidt and Lipson, 2009] propose an algorithm for finding meaningful invariances by learning to discard those invariances present in data that do not have a connection with physics.

Hamiltonian and Lagrangian equations are well-known formulations in conservative frameworks. Newtonian mechanics phrase problems in terms of constraints and forces, while Lagrangian and Hamiltonian formulations make use of energies and generalized coordinates. In addition, conservation laws can be easily derived from Noether's Theorem. This theorem, also known as Noether's first theorem, was developed in 1918 as a framework to explain how every symmetry in the system is naturally related to a conservation law. Symmetries are those transformations that leave a system invariant. Therefore, this invariance will reflect some kind of conservation. For instance, energy and momentum conservation can be derived from the time and space translation invariance respectively. A final advantage of this formulation is also the possibility to use powerful geometric techniques for studying the properties of dynamical systems.

Many works are dedicated to the study of Hamiltonian and Lagrangian descriptions to ensure energy conservation and symmetries emerging in this formulation in different settings in machine learning [Jin et al., 2020] [Wu et al., 2020] [Course et al., 2020] [Lei et al., 2016] [Jin et al., 2022] [Hesthaven et al., 2020] [Bertalan et al., 2019] [Greydanus et al., 2019]

[Toth et al., 2019] [Zhong et al., 2019] [Cranmer et al., 2020a]. Conversely, this framework is not applicable to problems of inherent dissipative nature that go beyond equilibrium.

The metriplectic formulation is a variation of the Hamiltonian (or symplectic) description to generalize its application to dissipative systems [Fish, 2005] [Nguyen and Turski, 2001]. They propose to complement the Poisson bracket of the Hamiltonian formulation $\{\cdot,\cdot\}$ with an additional symmetric bracket, called Leibniz or metric bracket $[\cdot,\cdot]$. Various examples of dynamical systems, such as fluid dynamics or plasma physics, are described by a metriplectic structure [Kraus, 2021]. This formalism is suitable for cases in which the conservative Hamiltonian description of a dynamical system includes unresolved degrees of freedom, that are not included in mesoscopic descriptions and that introduce dissipation by the fluctuation-dissipation theorem [Kubo, 1966]. Consider an initial microscale description at the molecular dynamics scale, that can be expressed in terms of a purely conservative formulation. The degrees of freedom, and therefore knowledge, that we omit growing from the micro to the meso and macro scales introduce dissipation that is included in the metric part of the formulation.

In this new framework, we describe the evolution of a functional $\mathcal{F}$ of a set of state variables z as

$$\frac{d\mathcal{F}}{dt} = \{\mathcal{F},\mathcal{G}\} + [\mathcal{F},\mathcal{G}]\,,$$

where $\mathcal{G} = \mathcal{H} - \mathcal{S}$ is a generalized free energy, $\mathcal{H}$ is the Hamiltonian or a generalization of the energy for the nondissipative contribution, and $\mathcal{S}$ an arbitrary function of the Casimir. A Casimir is an element that must not contribute to the reversible part of the dynamics. A metriplectic formulation can be defined with a different selection of potentials as long as the conditions of the metriplectic formulation are fulfilled.

2.5.1 GENERIC

GENERIC stands for General Equation for the Non-Equilibrium Reversible Irreversible Coupling [Grmela and Öttinger, 1997]. It is a metriplectic formalism that expresses the evolution of a system in terms of the so-called reversible (Hamiltonian potential) and irreversible (dissipative potential) contributions to describe the conservative and the dissipative parts of the dynamics of the system under study. It is a powerful tool for studying systems of practical interest since it guarantees 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.

In the case of GENERIC, these two contributions of the metriplectic formulation are described specifically with regard to the total energy E and entropy S of the system expressed of a set of state variables

$$\frac{d\boldsymbol{z}}{dt} = \boldsymbol{L}(\boldsymbol{z})\nabla E(\boldsymbol{z}) + \boldsymbol{M}(\boldsymbol{z})\nabla S(\boldsymbol{z}), \tag{2.1}$$

where $\boldsymbol{z}$ denotes a set of independent state variables that fully describe the thermodynamical state of the system. Without that information, we lack a GENERIC structure. Fluid dynamics are fully described in terms of the position and momentum of the particle discretization, internal energy, and, in the case of learning more complex fluids, the extra-stress tensor related to their microscopic evolution [Español et al., 1999]. $\boldsymbol{L}(\boldsymbol{z})$ is the Poisson matrix. It is antisymmetric and, together with the gradient of energy ∇E, characterizes the reversible part of the studied dynamics. $\boldsymbol{M}(\boldsymbol{z})$ is the friction matrix, which describes the dissipative irreversible characteristics of the system in conjunction with the entropy gradient ∇S. $\boldsymbol{M}$ is symmetric positive semidefinite.

GENERIC fulfills, by construction, the symmetries of the dynamical system that ensure the preservation of the symmetries related to the conservation and dissipation laws of thermodynamics by Noether's theorem. Hence, this structure-preserving formulation guarantees the conservation of critical quantities (mass, momentum), and the thermodynamical admissibility of the evolution of the system under study.

This formulation is supplemented by the degeneracy conditions to ensure the reversibility of the contribution of the Hamiltonian to the dynamics

$$\boldsymbol{L}\frac{\partial S}{\partial \boldsymbol{z}} = 0,$$

and that the entropy contribution cannot be affected by the operator that generates the reversible dynamics

$$\boldsymbol{M}\frac{\partial E}{\partial \boldsymbol{z}} = 0.$$

By imposing $\boldsymbol{L}$ to be skew symmetric, and $\boldsymbol{M}$ symmetric, positive semi-definite we guarantee that

$$\dot{E}(\boldsymbol{z}) = \nabla E(\boldsymbol{z}) \cdot \dot{\boldsymbol{z}} = \nabla E(\boldsymbol{z}) \cdot \boldsymbol{L}(\boldsymbol{z})\nabla E(z) + \nabla E(\boldsymbol{z}) \cdot \boldsymbol{M}\nabla S(\boldsymbol{z}) = 0, \tag{2.2}$$

to comply with the conservation of energy in closed systems, and

$$\dot{S}(\boldsymbol{z}) = \nabla S(\boldsymbol{z}) \cdot \dot{\boldsymbol{z}} = \nabla S(\boldsymbol{z}) \cdot \boldsymbol{L}(\boldsymbol{z})\nabla E(\boldsymbol{z}) + \nabla S(\boldsymbol{z}) \cdot \boldsymbol{M}\nabla S(\boldsymbol{z}) \geq 0. \tag{2.3}$$

which fulfills that entropy does not decrease and that it is generated in irreversible systems.

Given the measurements of the state of the system $\mathcal{S}$ at different discrete time steps, $\boldsymbol{Z}$, we could also obtain in discrete form the elements $\boldsymbol{L}$, $\boldsymbol{M}$, ∇E and ∇S of the GENERIC formalism, described in Eq. 2.1, by performing a regression process over the discretized expression of GENERIC

$$\frac{\boldsymbol{z}_{n+1} - \boldsymbol{z}_n}{\Delta t} = \mathsf{L}_n\mathsf{DE}_n + \mathsf{M}_n\mathsf{DS}_n. \tag{2.4}$$

Our work aims 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, but it could be unknown in the reduced-order manifold. As a final goal, we aim to construct the constitutive manifold of the sloshing dynamics. This concept described in [Ibáñez et al., 2018a] 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 slow manifold of $\{\mathsf{L}, \mathsf{M}, \mathsf{DE}, \mathsf{DS}\}$ 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:

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

as usual in the finite element community. To determine the numerical value of each GENERIC constituent, we accomplish a regression over discrete time intervals $\mathcal{J} \subset \mathcal{I}$,

$$\mu^* = \{\mathsf{L}_n, \mathsf{M}_n, \boldsymbol{A},_n \boldsymbol{B}_n\} = \arg\min_{\mu} \|\boldsymbol{z}(\mu) - \boldsymbol{z}^{\text{meas}}\|, \tag{2.5}$$

subject to the constraints:

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

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

which are the discrete form of the degeneracy conditions.

This formulation is also consistent in the low dimensional manifold we have built [Öttinger, 2015]. It will greatly reduce the computational cost of the optimization and the reconstruction of new, previously unseen, sloshing phenomena. L and M are squared matrices whose shape we frequently know from the description of the problem we model—there is a vast literature in the field [Romero, 2009b] [Portillo et al., 2017] [Ghnatios et al., 2019] [Mielke, 2011] [Grmela and Öttinger, 1997]. Nevertheless, they cannot be projected to the non-linear, low-dimensional manifold where the database has been projected, where we risk losing the rich thermodynamic structure induced by the degeneracy conditions. As a result, they are also the objective of the regression procedure in the reduced manifold.

2.6 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, we will analyze their efficacy to reach the best 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 chapter.

As mentioned earlier, the proof-of-concept problem that we have considered is discretized by a total number of 2898 SPH particles. 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 into the accuracy of the scheme.

2.6.1 Manifold learning evaluation

Analysis through POD

Proper Orthogonal Decomposition is the first approach to test. To apply this linear strategy, we have started by selecting equally spaced snapshots at intervals of $t = 0.02$ seconds. Additionally, the structure of the GENERIC matrix L is known [Vázquez-Quesada et al., 2009],

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

so only the discrete matrix form of the gradients A and 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 show a typical pattern. In it, see Fig. 2.5, 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 the energy of the system is captured. By increasing the number of modes selected, we reach 77.43% of energy captured with 50 modes, which 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. When analyzing the eigenvalues of the problem we can appreciate that, although some modes 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 decide to use 25 modes to perform a reasonable calculation and generate a credible representation of the fluid for each trajectory. Notwithstanding the mentioned difficulties—not surprising, given the linearity of the POD method—the L_2 norm error in the reconstruction of the velocity field, see Fig. 2.6, 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 [Romero, 2009a]—but to the big number of

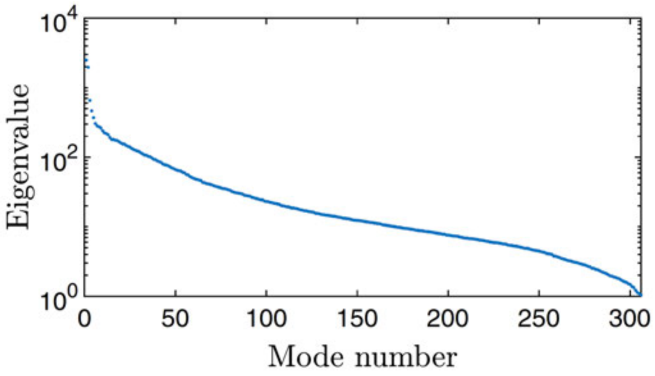


Figure 2.5: Evolution of the eigenvalues of the pseudo-experimental results in POD approach.

POD modes necessary to obtain an accurate representation of the non-linear character of the pseudo-experimental results.

Analysis through LLE

Through the application of LLE, a lower number of modes is expected to capture the essential dynamics and obtain a consistent 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 compared to the POD approach. This error resulted to be on the order of 0.017%. The reconstruction of the displacement in time of a specific particle (number 1) is shown in Fig. 2.7.

Analysis through TDA

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 $t = 0.00272$ seconds. Each trajectory is successfully embedded in a manifold of $d = 3$ dimensions for calculating the set of $\{\mathsf{M}_n, \boldsymbol{A}_n, \boldsymbol{B}_n\}$ matrices.

Remark 1 The set of constituents of the GENERIC expression of the problem cannot, 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 losing the properties of these terms 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. 2.5 is accomplished as

$$\mu^* = \{\mathsf{M}_n, \boldsymbol{A}_n, \boldsymbol{B}_n\} = \arg\min_{\mu} \|\boldsymbol{x}_\mu - \boldsymbol{x}^{\text{meas}}\|\,,$$

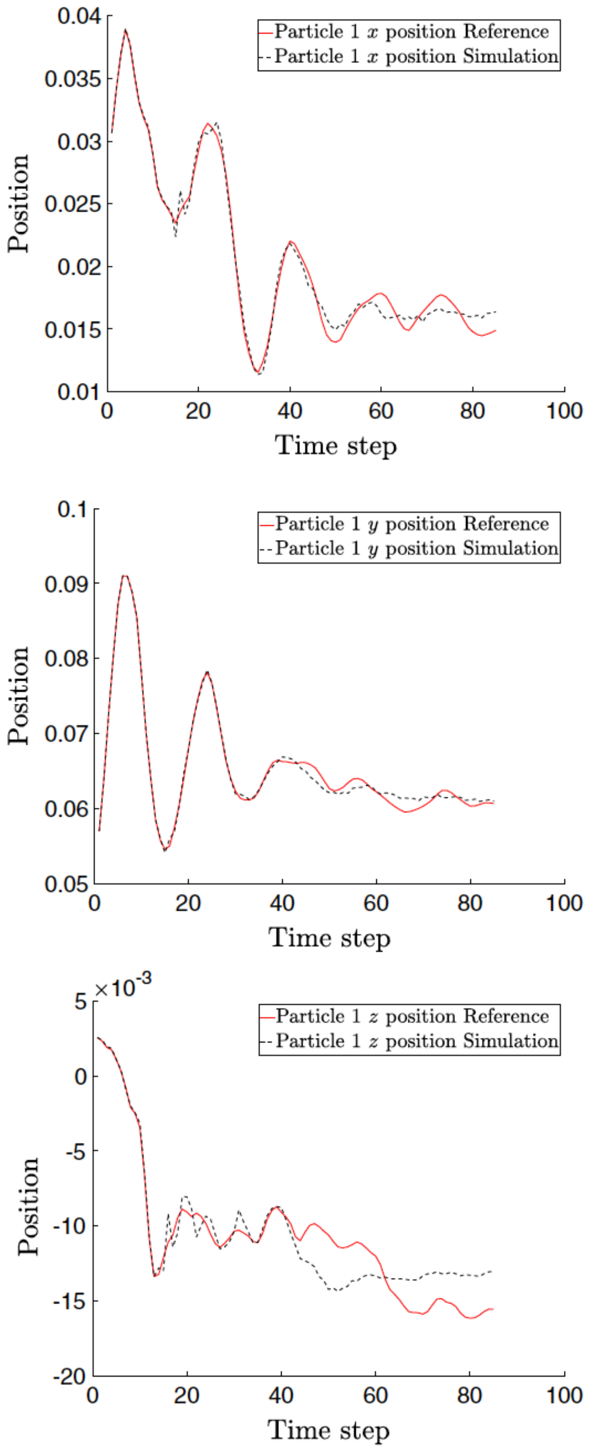


Figure 2.6: POD-based reconstruction of the trajectory of one particular particle (evolution of coordinates x, y and z along time) vs. pseudoexperimental results. Initial velocity 0.15 m/s.

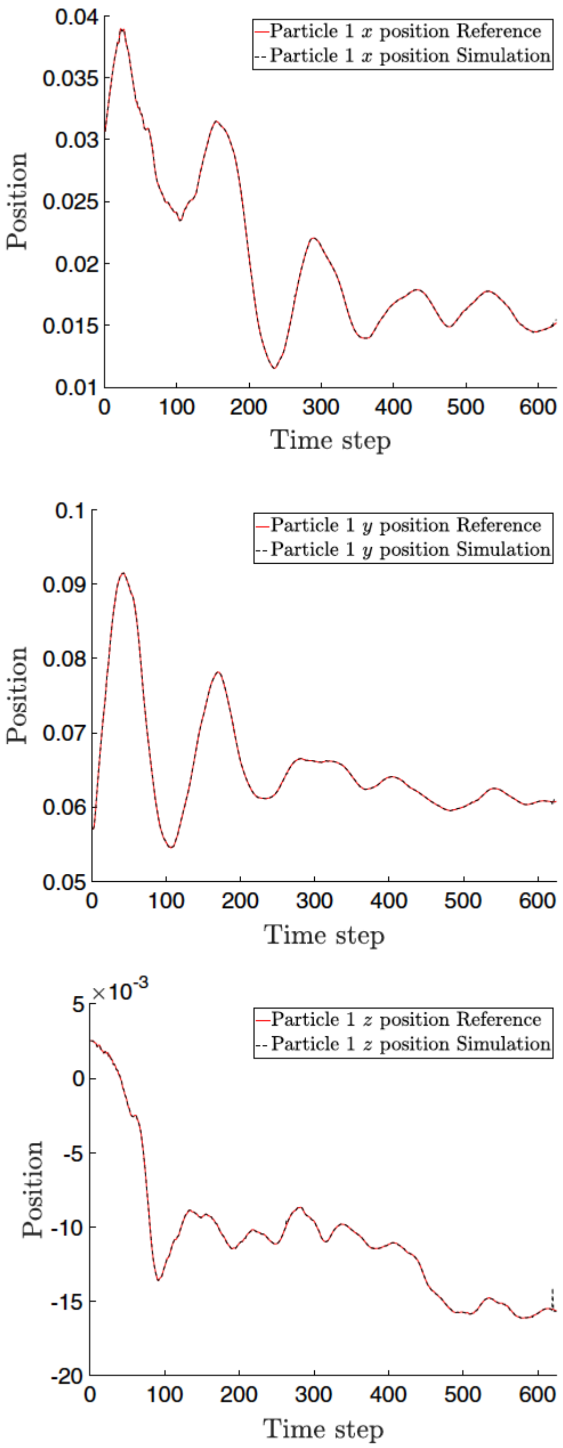


Figure 2.7: LLE-based reconstruction of the trajectory of one particular particle (evolution of coordinates x, y and z along time) vs. pseudoexperimental results. Initial velocity 0.15 m/s.

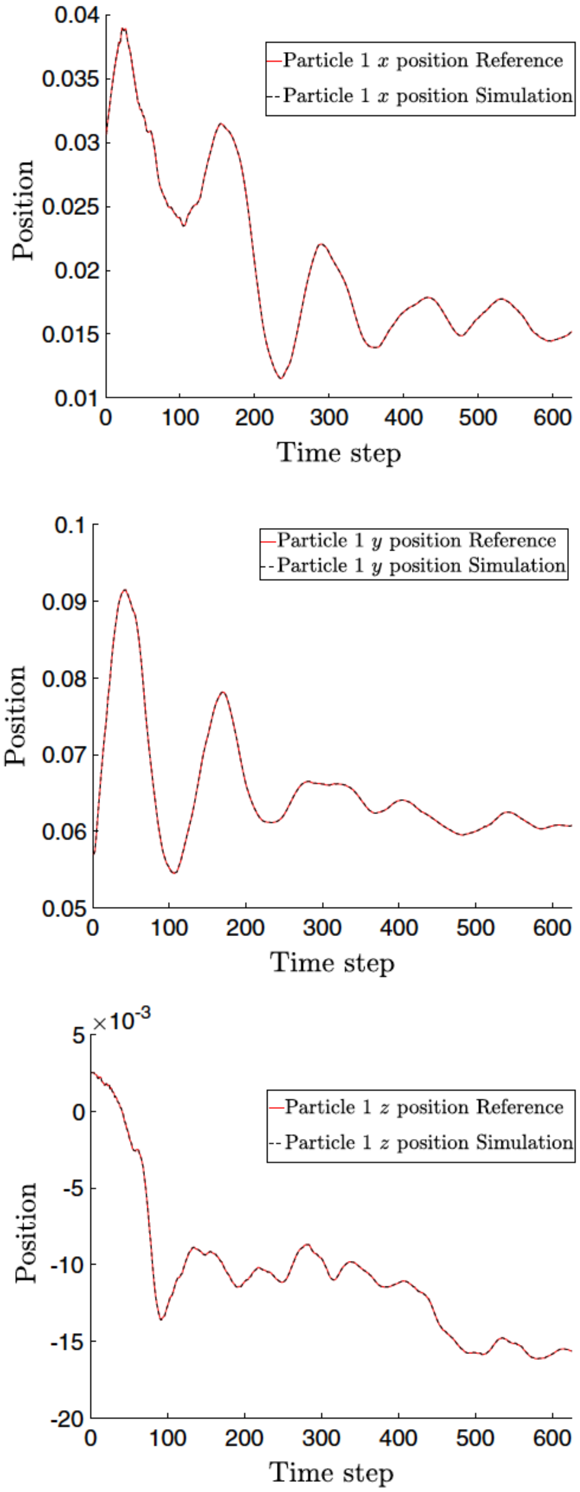


Figure 2.8: TDA-based reconstruction of the trajectory of one particular particle (evolution of coordinates x, y and z along time) vs. pseudoexperimental results. Initial velocity 0.15 m/s.

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. This error is represented in the reconstruction of the time evolution of one particle in Fig. 2.8. This very low error justifies the selection of the TDA-based approach for a general case, whose details will be analyzed next.

2.6.2 Integration in the reduced-order manifold

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. We have previously considered four simulations to create the training dataset for learning the constitutive manifold of the sloshing. In the next phase, we use a test simulation out of the training dataset to evaluate the performance of the learned simulation. We provide the first snapshot and the simulator engine has to recover the time evolution of the dynamics.

To test the integration scheme's efficacy, we will 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. 2.4—recall also Fig. 2.3—we obtain a scheme of the form

$$\boldsymbol{x}_{n+1} = \boldsymbol{x}_n + \Delta t \left[\mathsf{L}_n \boldsymbol{A}_n + \mathsf{M}_n \boldsymbol{B}_n \right] \boldsymbol{x}_n,$$

where we have highlighted the dependence of every term of the GENERIC description of the movement on n, if an explicit scheme is chosen. Since no pseudo-experimental results for n will exist, these values must generally 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. 2.9. Note the visible similarity between the results obtained by the proposed method and those obtained by SPH.

In Fig. 2.10 we show the reconstruction of the displacement of particle number with label 1. Of course, the error increases compared to the examples in the preceding sections, since there is no experimental result that coincides with the simulated problem. The relative mean-squared error evolution in time is plotted in Fig. 2.11. It is worth noting that the error remains under 2% in the vast majority of the time increments. Only in a small portion of the increments does the error rise to a value always under 10%. Of course, further refinement in the data sampling will produce more accurate results.

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

2.7 Conclusions

What we have developed in this chapter is the foundation for a practical way of learning the behavior of a free-surface fluid that allows overcoming 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 the fluid without losing information on the displacement and velocity fields based on the GENERIC formulation.

GENERIC has several appealing features. First, its terms can be obtained numerically by regression of available experimental data. This approach guarantees the correct satisfaction of energy conservation and entropy dissipation. Second, numerical discretization of GENERIC by finite differences in time provides us with a powerful and consistent integration scheme that has shown to possess remarkable numerical properties, as previously studied by different authors. The third ingredient of the method is nonlinear model order reduction.

Since GENERIC establishes a system of equations for every particle in the model and given the fact that this number is usually very high, we reduced the dimensionality 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 can cope with severe real-time constraints while maintaining its desired thermodynamic consistency. Results showed that the simulations ran slightly faster than in 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. Provided the obtained results for emulation of water behavior, we consider its application to a wider range of fluids of Newtonian and Non-Newtonian nature and the implementation of a digital twin for online learned simulation.

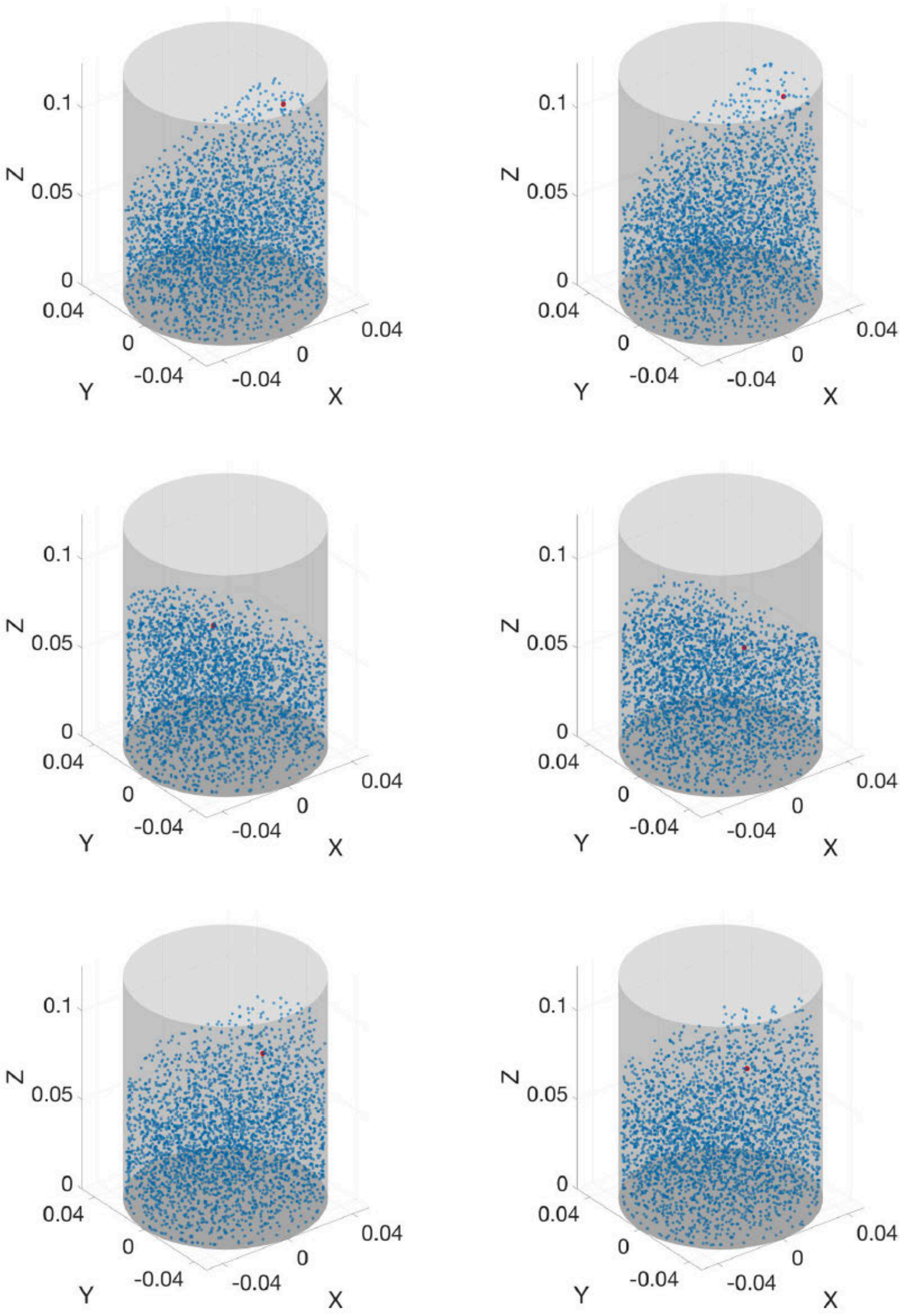


Figure 2.9: Comparison between snapshots obtained by the proposed method (left column) and their equivalent ground truth obtained by SPH (right). Time instants 28, 115 and 172 are shown. Particle 1 is highlighted in red so as to ease the comparison among pseudo-experimental and learned results.

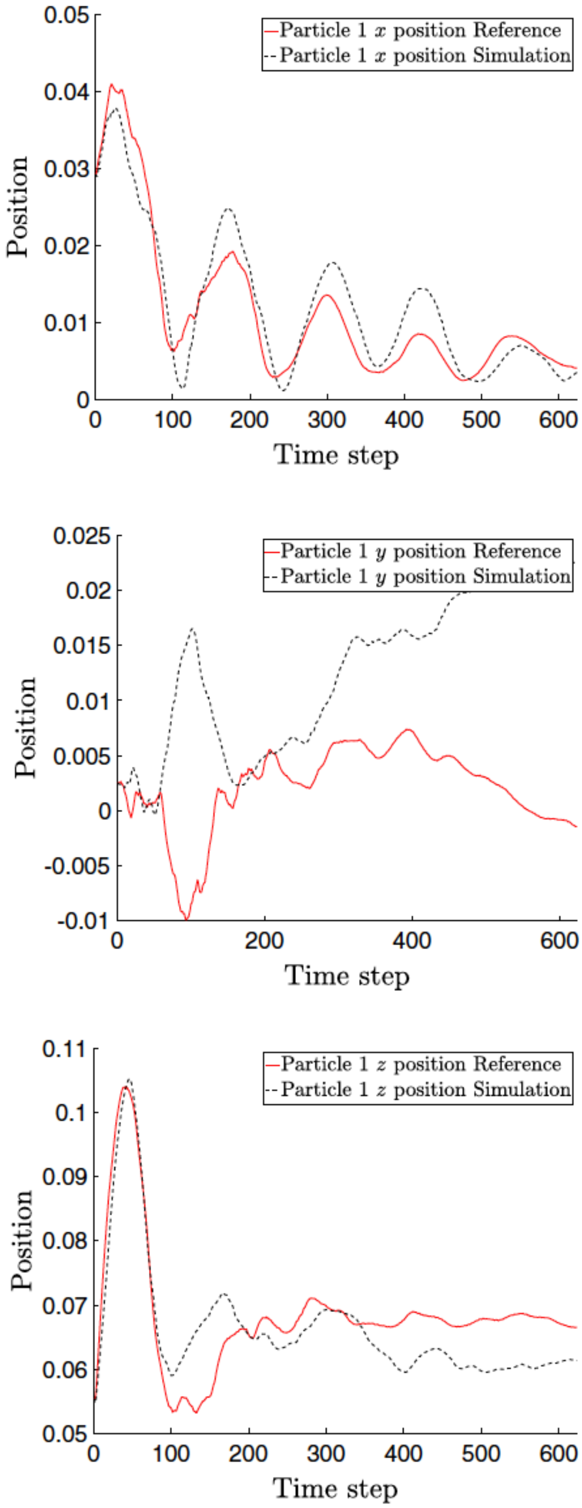


Figure 2.10: Integration of the sloshing movement versus ground truth. Initial velocity 0.175 m/s. Fig.s represent, respectively from top to bottom, the displacement of particle with label 1 along x, y and z coordinates.

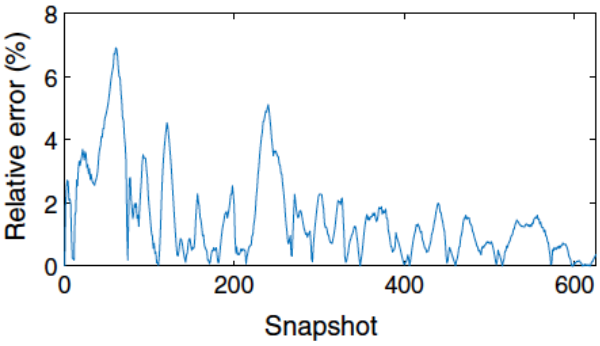


Figure 2.11: Evolution in time of the error in the prediction of the water height.

Chapter 3

Developing physically sound, self-learning digital twins for sloshing fluids

In this chapter, a self-learning digital twin strategy is developed for fluid sloshing phenomena. The proposed method infers the (linear or non-linear) constitutive behavior of the fluid from video sequences of the sloshing. Real-time prediction of the fluid response is obtained from a reduced-order model (ROM) constructed using our thermodynamics-informed data-driven learning approach. From these data, we aim to predict the future response of a twin fluid reacting to the movement of the real container. The constructed system can perform accurate forecasts of its future reactions to the movements of the glass. The system is completed with augmented reality techniques to enable comparisons among the predicted result with the actual response of the same liquid and to provide the user with insightful information about the physics taking place. The content of this chapter is included in the following publication:

- Moya, B., Alfaro, I., González, D., Chinesta, F., & Cueto, E. (2020). Physically sound, self-learning digital twins for sloshing fluids. PloS one, 15(6), e0234569.

3.1 Introduction

Considering the results obtained in the development of a simulation engine for water, we take a step forward in the implementation of learned simulation from observations. For this purpose, we assemble the digital twin and broaden the spectrum of considered fluids. As a result, we face new challenges in the implementation. The fluids employed in the new descriptions will be both Newtonian and non-Newtonian, and the simulation must learn correctly the behavior of all. In addition, we are still restricted by the real-time constraints, now also in the data-acquisition process and loop performance. Finally, a tool is required for distinguishing fluids and employing the appropriate model for the detected liquid.

Digital twins are tools to fully address real-world phenomena and seamlessly connect reality with virtual predictions. The abundance of data inputs allows relating physical objects or processes with the simulation to obtain online meaningful results. The digital twin of the object of interest offers, in the end, a cost-effective real-time solution for making decisions analogous to those that we would make interacting with the real process or object.

These virtual replicas have already shown their usefulness through their implementation across many kinds of industries and branches of science. In the field of robotics, they are the means for scene understanding and interaction to connect robots with the actual reality to develop a method for physical scene understanding in real-time and with real measurements [Pan et al., 2016] [Schenck and Fox, 2018a].

While finite elements (or related numerical simulation techniques) are primarily used for offline product development, design, and simulation, real-time digital twins offer applications for continuously changing, monitored systems [Chinesta et al., 2018]. The development of the latter is more infrequent than the former due to their highly demanding computational cost. Reduced-order models supply with tools to face such disadvantage [Badías et al., 2018] [Aversano et al., 2019] [Keiper et al., 2018] [Tezzele et al., 2018].

[Kapteyn et al., 2020] face difficulties that show some similarities to our problem. They work on physically-constrained digital twins, developed in a reduced-order space. Their particular object of study is aircraft replanning, and propose a library of models with regard to the detected conditions. They identify the proper model that will fit its behavior best based on a tree classification with the data available. The model for each topology is defined on a reduced basis, giving support for immediate decision-making.

In this chapter, we present the implementation of a novel self-learning digital twin of fluid sloshing. By self-learning digital twin we mean a system that, rather than performing data assimilation (to determine the viscosity of the fluid, for instance) can construct a physically correct, data-driven replica of a previously unseen fluid, regardless of its constitutive behavior. The digital twin employs computer vision to obtain data from the real system. In particular, in this application, we track the container of the fluid, which triggers the slosh. From these data, we aim to predict the evolution of the fluid dynamics in time. The system is complemented with augmented reality techniques, to enable comparisons among the predicted result with the actual response of the same liquid.

Augmented reality (AR) enables the contextualization of the twin in the physical world. Consequently, the interaction between reality and the replica, as well as user control and interpretation, are straightforward. AR tools have also been implemented for non-rigid representation, such as aerodynamics [Badías et al., 2019], or deformable objects [Badías et al., 2018]. AR applications of sloshing dynamics are not common, and they present limitations such as computing time, and localization in the scene, made with markers or features in the image [Fujisawa and Kato, 2009].

The workflow of the digital twin follows the next sequence. First, by employing computer vision techniques, we obtain data about the movement of the glass. Secondly, we feed the simulator engine with that information. Finally, the output is represented within a real container, which may be the one containing the fluid or not. We let the simulation run in the background to interpret the physics taking place.

3.2 Learning Newtonian and non-Newtonian fluids

We take a step forward from the simplified geometry of Chapter 2 to model the glass used to contain the real liquid entity of the digital twin. It has a slight slope that increases its diameter which can be seen in Fig. 3.1. In our proof-of-concept, the height of the glass is $h = 8$ cm and the bottom and top diameters are $b1 = 6.5$ cm and $b2 = 8$ cm, respectively. We consider a fluid volume adapted to the shape of the glass, filled up to 6 cm. Equally to the work already presented, data are obtained with the Smooth Particle Hydrodynamics theory [Monaghan, 1992].

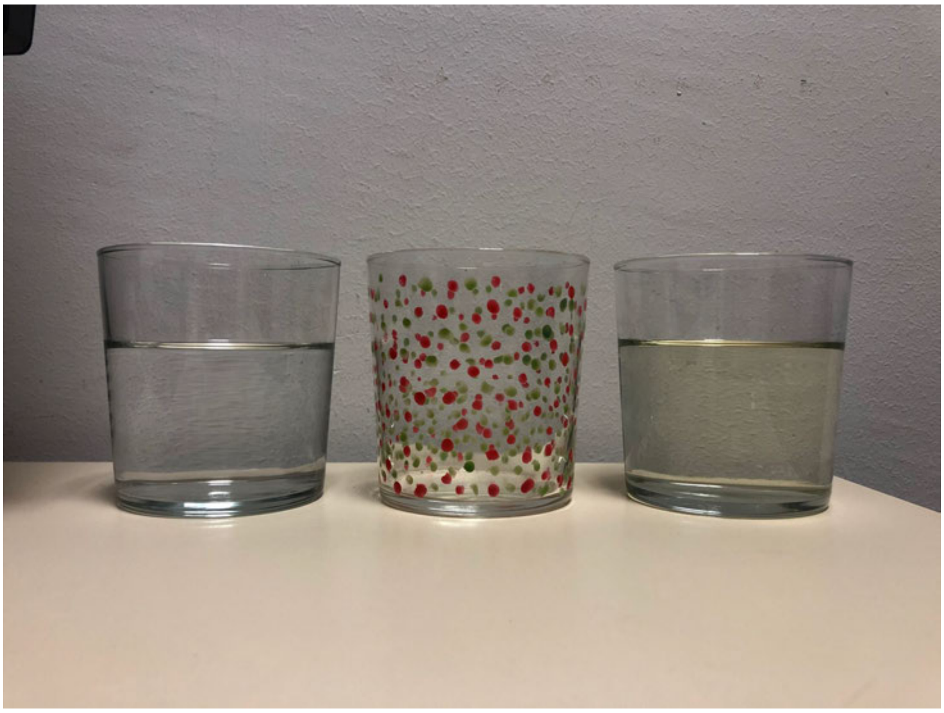


Figure 3.1: Containers used for simulation, representation and comparison with the virtual phantom. The three glasses are identical. The middle glass will be virtually filled with the replica of the liquid volume. Since it is empty, the painted texture helps the cameras to track its movement.

Let $j = 1, \ldots, M$ be the particle number, with which the fluid volume has been discretized. Although to describe a Newtonian behavior we only need their position $\boldsymbol{q}_j$, velocity $\boldsymbol{v}_j$ and internal energy E_j at discrete time instants, non-Newtonian fluids are not fully described with these state variables [Español et al., 1999]. Some of the liquids we attempt to describe here have viscoelastic properties, and this behavior is captured by including the extra stress tensor $\boldsymbol{\tau}_j$—or a related magnitude—of each particle in the set of state variables. The fluid is then characterized by a vector of these particle variables, such as

$$\mathcal{S} = \{\boldsymbol{z} = (\boldsymbol{q}_j, \boldsymbol{v}_j, E_j, \boldsymbol{\tau}_j) \in (\mathbb{R}^3 \times \mathbb{R}^3 \times \mathbb{R} \times \mathbb{R}^6), j = 1, 2, \dots, M\},$$

for every $n > 1$ equally spaced time steps in the interval $(0, T]$ of each simulation. The fluid volume has been discretized into $M = 2134$ particles. If we have 13 degrees of freedom per particle, the full dimensionality of each snapshot vector is $D = 27742$.

In this chapter, we generalize this approach to be able to identify any kind of viscoelastic fluid from data. A database has been constructed with pseudo-experimental results for glycerine, butter, honey, mayonnaise, and chocolate. We intend to cover different viscosities and densities, as well as behaviors. Blood, chocolate, and mayonnaise have been described as non-Newtonian fluids. Blood is usually Newtonian in this type of simulation, but we have decided to employ a non-Newtonian approximation for testing the method. Our goal is to prove the generalization of this method over liquids with different viscosities and densities. Blood, chocolate, and mayonnaise have been described as non-Newtonian fluids. The adaptation to different physics will set the bases to perform correction and adaptation to any new perceived fluid. The different fluid constitutive models in a rheogram are sketched in Fig. 3.2.

Chocolate, mayonnaise and blood[1] are shear-thinning [Izidoro et al., 2007] [Kumbár et al., 2018]. We have selected the Herschel-Bulkley model to reproduce their rheology [Herschel and Bulkley, 1926] . It follows the next expression

$$\tau(t) = k\dot{\gamma}^n(t) + \tau_0,$$

in which τ is the shear stress, k the consistency index, $\dot{\gamma}$ the shear rate, n the flow index, and τ_0 the yield stress. The flow index indicates the behavior of the fluid, being $n < 1$ for the shear-thinning, and $n > 1$ for the shear-thickening like the oobleck. The consistency index is a proportionality constant and can show approximately the magnitude of the viscosity when comparing fluids with similar n.

For all these fluids, synthetic data have been obtained according to the parameters in Table 3.1.

We have simulated four different sloshing trajectories for each liquid. It is also worth mentioning that the stable time step of the computation was defined as 0.001 seconds. Despite having all these data, we will select equally spaced snapshots, pruning the database to avoid overfitting and undesirable noise of the synthetic measurements.

Our digital twin needs to produce results at a real-time frequency, which is 30 Hz. Therefore, our constraint is to select a time step of at least 0.03 seconds. Nevertheless, the

[1] https://www.michael-smith-engineerscouk/mse/uploads/resources/useful-info/General-Info/Viscosities-of-Common-Liquidspdf

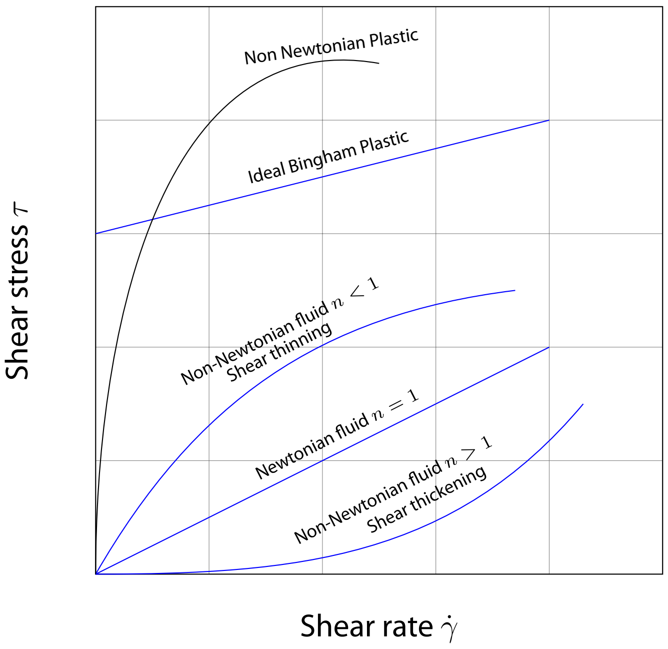


Figure 3.2: Standard classification of the fluid families considered herein. In the case of Newtonian fluids, their properties are constant over time and show a linear response. Their flow index n is then set to 1, and their yield stress to $\tau_0 = 0$. In contrast, shear thickening fluids start flowing when the stimulus is greater than the yield stress $\tau_0 > 0$. For these fluids, $n > 1$. In this work, fluids whose behavior can be assimilated to shear-thinning. Fluids that incorporate some kind of plastic behavior need special treatment and have not been considered yet.

Fluids	k (Pa· s)	n	τ_0 (Pa)
Glycerine	0.95	1	-
Blood	0.017	0.708	-
Mayonnaise	45.40	0.495	98.18
Melted chocolate	5.764	0.697	9.096

Table 3.1: Characteristics of the fluids considered in this work.

selected time step for the learning process must ensure that we capture all the important features of the dynamics. If we considered a larger time step, we might miss important details of the phenomenon.

3.3 Manifold learning of fluids: the k-PCA

Our strategy here is to assume that the already discretized sloshing dynamics of each fluid evolve on a finite-dimensional, smooth, and real manifold $\mathcal{M} \in \mathbb{R}^D$. This manifold is reconstructed from the synthetic data previously obtained.

We make use of model order reduction techniques to find a reduced-order manifold $\mathcal{N} \in \mathbb{R}^d$, where $d \ll D$, to embed the manifold to achieve real-time performance for our digital twin. Again, Fig. 2.3 sketches our approach. On this reduced-dimensionality manifold, we will preserve the important features of the dynamics expressed in a new system of latent variables. Despite its lack of physical meaning, they capture the patterns of the dynamics from the dataset. These techniques will provide insights about the manifold where all the fluids are embedded to study their correlations.

The dominant non-linear and high dimensional nature of the problem has shown difficulties to be embedded in a manifold led by linear correlations [Moya et al., 2019]. Instead, it showed a good performance with non-linear model order reduction techniques (namely, LLE [Roweis and Saul, 2000] and TDA [Wasserman, 2018]). Following these results, we have employed k-PCA (kernel Principal Component Analysis) to distill the reduced-order manifold of each fluid [Schölkopf et al., 1998]. Although TDA provided accurate results, the computational resources required for the method posed a problem for the implementation of the twin. k-PCA overcomes these limitations to build a real-time digital twin at a lower computational cost.

Given a matrix $\boldsymbol{Z}$ of fluid snapshots, we compute the product $\boldsymbol{S} = \boldsymbol{Z}\boldsymbol{Z}^T$ to obtain the matrix of pairwise scalar products. The key hypothesis of k-PCA is that the projection of the points to a new space $\phi : \mathcal{M} \subset \mathbb{R}^D \to \mathbb{R}^Q$, where Q is the new dimension, probably higher than the current space, can result to be linearly separable. Then, we apply PCA in $\mathbb{R}^Q$. As a result, we obtain the most relevant nonlinear principal components of $\boldsymbol{Z}$ and thus a projection to a much lower-dimensional manifold.

In our problem, we have a matrix $\boldsymbol{Z}$ of snapshots. Each column $\boldsymbol{z}_i$, $i = 1, \ldots, n$ is a snapshot, a vector of state variables that represent the state of the fluid at a specific time instant. For each fluid, we have a total of n snapshots,

$$\begin{bmatrix} | & | & & | \\ \boldsymbol{z}_1 & \boldsymbol{z}_2 & \cdots & \boldsymbol{z}_n \\ | & | & & | \end{bmatrix} = \boldsymbol{Z} \in \mathbb{R}^{D \times n}.$$

With this method we successfully embedded the points of each fluid into a manifold $\mathcal{N} \subseteq \mathbb{R}^{d=3}$, see the results in Section 3.8.

3.4 Dynamics reconstruction from acquired data

The digital twin is designed to learn the description of the dynamics of the fluids, modeled as a time-evolution problem, expressed as a function of a set of variables

$$\boldsymbol{z}_t = \boldsymbol{z}(t) \in \mathcal{S} \subset \mathbb{R}^D,$$

with D representing the full dimensionality of the problem. These variables are those required for a thermodynamically admissible representation of the sloshing dynamics, evaluated at each particle in which the fluid has been discretized. Note that some of these variables are, in general, not all measurable with a camera. Then, we must develop a link between the state of the fluid and the information that we obtain via computer vision techniques. In this case, we have related the velocity of the glass that causes the sloshing effect with the state variables under this condition. As a result, scene information becomes an interpretable input for our integration scheme.

The physical constraints that we impose have the ultimate goal of developing more accurate and generalizable models. Despite the simplification of the calculation method, those models still entail a high computational cost difficult to surmount that is overcome with model order reduction to predict the next state of the fluid at a frequency of 30 Hz or faster (the one at which standard cameras operate).

This time, we apply GENERIC learning to coarsely model the macroscopic behavior of fluids with different densities and viscosities, finding patterns that root at the micro-scale level. Some of these fluids are non-Newtonian and show viscoelastic properties. We require a more complex formulation that is to be learned from the new selection of state variables. With these measurements obtained from computational simulations, we perform the optimization over the discretized friction matrix M, and the approximation of the gradients of energy and entropy, as described in Chapter 2, considering L to be known in the regression

$$\boldsymbol{\mu}^* = \{\mathsf{M}, \boldsymbol{A}, \boldsymbol{B}\} = \underset{\boldsymbol{\mu}}{\text{argmin}} \, ||\boldsymbol{z}(\boldsymbol{\mu}) - \boldsymbol{z}^{\text{meas}}||, \tag{3.1}$$

including the discretized degeneracy conditions in the optimization.

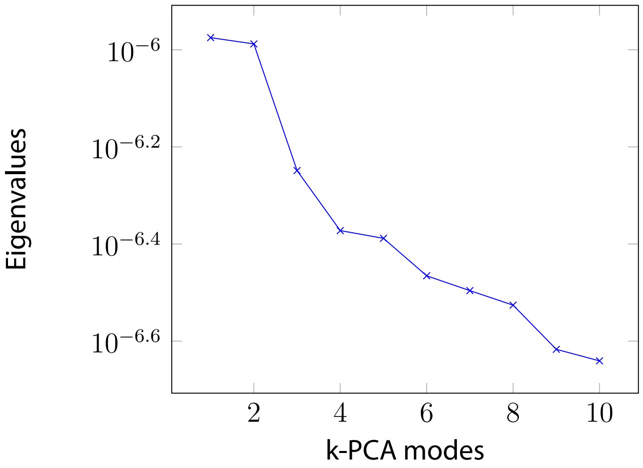


Figure 3.3: This picture represents the evolution of the eigenvalues for the first 10 k-PCA modes for the whole set of fluids. We distinguish three modes that stand out over the others. This fact justifies the reduction performed. As a result, we aim to provide a more manageable and efficient system for classification.

3.5 Fluid clustering and classification

The digital twin should be able to mimic the behavior of any type of fluid, not necessarily in the database. Then, in future applications, it must perceive and recognize the fluid in the scene to perform accurate calculations. In other words, it must interpret data from the scene to explain what it is watching in understandable terms for a human.

We expect to classify the fluid with only a few observations and, knowing which fluid it is—or interpolating between the closest neighbors in the phase space—, make the calculations in the appropriately reduced-order manifold.

We remind that the dimensionality of the problem is of order $D \sim 10^4$. This would make it unfeasible to perform the classification as fast as the application requires. We address this problem by applying model order reduction techniques to perform the classification in a manifold of much lower order. Dimensionality reduction is a common preprocessing step for classification tasks [Wang and Carreira-Perpinan, 2014]. The time invested in learning to map the dataset to a reduced space is compensated with the time reduction entailed in the classification itself. In addition, dimensionality reduction can improve the efficacy of the classification. If the dataset has a lower-dimensional structure, we avoid the noise of the large dimensional set, which results in an improvement in the accuracy.

k-PCA enables us to project the data onto a low dimensional manifold, of 3 dimensions in our case (see the eigenvalue evolution in Fig. 3.3). The different types of fluids remain clustered in the new projection, as can be seen in Fig. 3.4. k-PCA can unveil the features that make the behavior of each fluid unique with respect to the others. We expect this fact to be advantageous for the classification process.

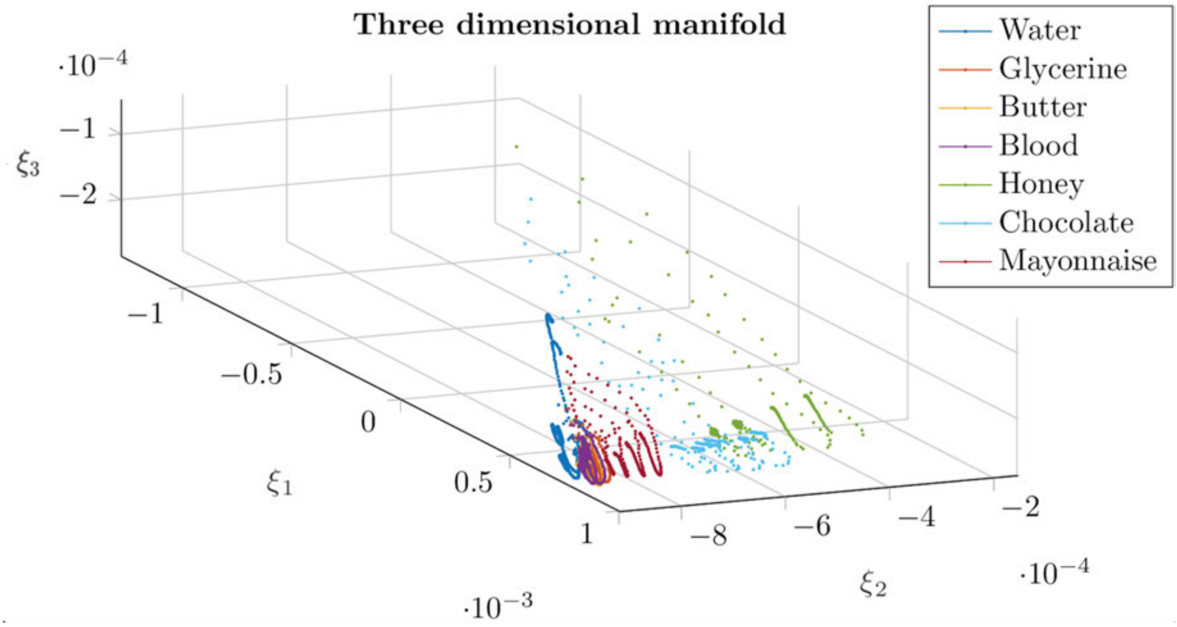


Figure 3.4: By employing k-PCA we reach a manifold of 3 dimensions where the different fluids, represented by one color each, remain clustered.

In our case, we employ random forests for classifying our dataset [Breiman, 2001]. This technique consists in constructing several decision trees in different subspaces of the training set to generalize the classification and, as a result, avoid the overfitting that usually appears in single regression tree techniques.

We also apply k-fold cross-validation as part of the learning process. The method suggests splitting the database into two parts, one assigned for training and the rest saved for testing. This process is repeated iteratively k loops, changing the distribution data of training and test sets, to improve the fitting.

We need enough data to recognize the underlying trend, but we also need to leave sufficient for testing to avoid high variance error. According to this criterium, we establish a relation of 80% of snapshots for training, and 20% for testing, in our algorithm. We trained the model following a cross-validation scheme in $k = 5$ iterations.

Overall, the results obtained from the classification algorithm showed a good performance. High accuracy was achieved by analyzing both global results as well as the error obtained for each fluid individually (Table 3.2). With this result, we consider the model valid for our problem, as well as for decision-making applications.

3.6 Observation: the camera model

Our physics perception is mainly developed from visual inputs. Hence, computer vision is the experimental technique to collect real-world data. Images represent shapes of people

Individual Accuracy	Water	99.17%
	Glycerine	97.32%
	Butter	92.88%
	Blood	91.07%
	Honey	91.07%
	Mayonaise	91.07%
	Chocolate	91.07%
Global Accuracy	95.93%	
Precission	99.17%	
Recall	99.17%	

Table 3.2: Results obtained from classification after random forest training with k-fold cross validation. Global accuracy is over 95%, and individually all remain over 90%.

and things captured by an eye, a mirror, an optic device, or a camera due to the reflected light. As a result, we obtain a visual representation that we store as a source of information. Interpretation of the information of visual inputs is a key point in our daily routines. For example, stereopsis, known as the visual ability to see in three dimensions, makes us able to perceive depth in the surroundings and interact with the environment. An appropriate 3D reconstruction contributes to a high quality of life and the understanding of reality. Equally, there are some other human capacities related to visual inputs. It is because of this information that we can distinguish objects, learn patterns, and enrich our cognition and reasoning.

Computer vision is the branch of artificial intelligence in charge of studying the sense of vision to replicate these interpretation capacities. This field has experienced a flourishing decade where many applications have been developed in pattern analysis, tracking, and scene reconstruction. In addition, several systems for physical scene understanding make use of these methodologies [Schenck and Fox, 2016a] [Schenck and Fox, 2018a] [Nava et al., 2021] [Yan et al., 2020].

Computer vision foundation is the pinhole camera model, represented in Fig. 3.5, which consists in an approximation to correlate a point in the three-dimensional space with its projection onto a 2D image plane considering that the rays are projected from a common center of projection. As it is an ideal model, it does not consider distortions, blur, and other effects due to the characteristics of the lenses and the aperture of the camera. These are taken into account usually in a previous rectification step.

We consider a point in three-dimensional world coordinates $(X, Y, Z) \in \mathbb{R}^3$ expressed as $\boldsymbol{p}_w = (X, Y, Z, 1)$. This point has a projection in an image 2D plane $(u, v) \in \mathbb{R}^2$ represented as $m(u, v, 1)$. The pinhole model enables the definition of a mapping $\Pi : \mathbb{R}^3 \to \mathbb{R}^2$ for this projection. If the 3D point is referenced to the local coordinate system of the cam-

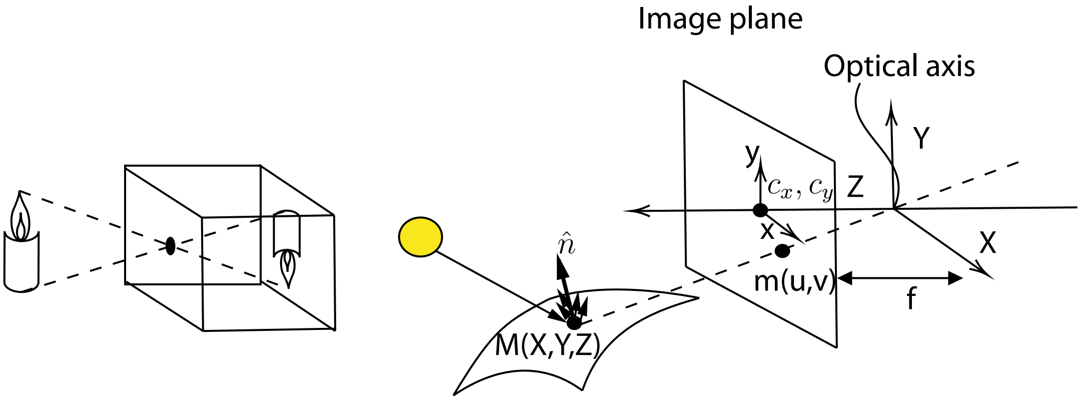


Figure 3.5: Scheme of the pinhole model. It represents the projection of the 3D world onto 2D images.

era, the projection to the image plane is defined as:

$$\begin{pmatrix} u \\ v \end{pmatrix} = f \begin{pmatrix} X/Z \\ Y/Z \end{pmatrix}.$$

Nevertheless, in a situation with multiple cameras and changing positions, the 3D position of any point must be referenced to a global coordinate system that can be observed by every camera. In this situation, the relationship between the 3D world and a 2D image, and vice versa, is built from the so-called intrinsic and extrinsic parameters. Intrinsic parameters relate the 2D position of a point, in pixel coordinates, with its 3D position to the camera reference. Those parameters are the pixel spacing s_x and s_y, the center coordinates of the camera (c_x, c_y), and focal length in x direction, $f_x = f/s_x$, and y direction, $f_y = f/s_y$, directions. They form the calibration matrix $\boldsymbol{K}$.

On the other hand, extrinsic parameters are those that represent the camera's rotation ($\boldsymbol{R}$) and translation ($\boldsymbol{t}$) around a reference coordinate system. With all this information, a point can be mapped from the real world to the pixel coordinates, and vice versa.

Having prior knowledge of the camera's calibration and position, the 3D estimation of every point is performed through *triangulation* [Hartley and Zisserman, 2003]. As mentioned before, the camera outputs the intrinsic and extrinsic parameters at each frame. As a result, we know the projection of the 2D features in the 3D real-world system to perform any operation in the algorithm implemented,

$$s \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix},$$

$$\tilde{\boldsymbol{x}}_s = \boldsymbol{K}[\boldsymbol{R}|\boldsymbol{t}]\boldsymbol{p}_w.$$

The projection of a 3D point to an image is straightforward given the intrinsic and extrinsic parameters of the camera for a known 3D position. However, there are situations where this information is partially or completely unknown. Given a 3D point observed by two cameras, whose position and calibration are known, its 3D position can be reconstructed through multi-view triangulation. In addition, bundle adjustment, whose scheme is shown in Fig. 3.6 is a particular optimization algorithm used to solve this optimization problem when the position of the cameras is also unknown. Structure-from-motion is the problem of reconstructing a 3D scene in the latter exposed conditions, and a bundle adjustment is a standard algorithm for its resolution. Given a point cloud of matching points, detected by a specific algorithm—see SIFT, for instance [Lowe, 1999]—, the optimization will try to minimize the positions of the points as well as the rotation $\boldsymbol{R}$ and translation $\boldsymbol{t}$ matrices.

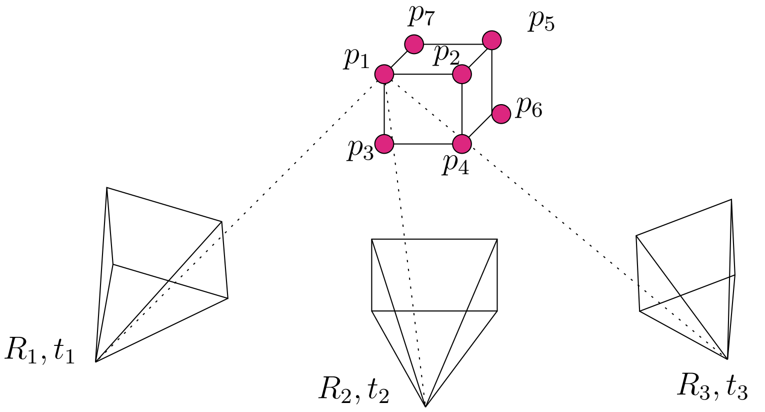


Figure 3.6: Bundle adjustment. Three cameras, whose rotation and translation information are unknown, capture the same features of an object to perform the optimization over the unknown data and perform the estimation of the cameras' position and 3D coordinates of the features of the object.

Some online applications require performing this estimation on-the-fly at the same time the application is running. This problem is known as Simultaneous Localization And Mapping (SLAM) [Durrant-Whyte and Bailey, 2006]. This localization is proposed to be made based on sensored information [Gutierrez-Gomez et al., 2016] or feature extraction [Mur-Artal et al., 2015], to name a few. In this discipline, there is a sub-field called Non-Rigid structure from motion, which stands for situations where the objects in the scene also move and deform. In this type of environments [Recasens et al., 2021] [Badias et al., 2021], as it happens with fluid, positioning and feature extraction becomes complicated. However, the cameras employed in this work provide the intrinsic and extrinsic parameters of the camera. In addition, it will remain in a fixed position.

The present work has been developed using the stereo camera Zed Mini model from Stereo Labs (`https://www.stereolabs.com/zed-mini/`). This camera incorporates a stereo system and an Inertial Measurement Unit, IMU. The camera can instantaneously provide the user with its intrinsic $\boldsymbol{K}$ and extrinsic, $\boldsymbol{R}$ and $\boldsymbol{t}$, parameters. This fact helps to speed up the computation of the inputs and outputs of our digital twin, as well as its augmented reality reconstruction.

3.7 Data extraction from the scene

Next, we analyze data extraction with computer vision techniques to understand how we perform scene understanding and connection with the twin. We place the camera in front of the glass at a fixed position to evaluate in real-time the movement of the glass and the free surface evolution of the liquid, as shown in Fig. 3.7.

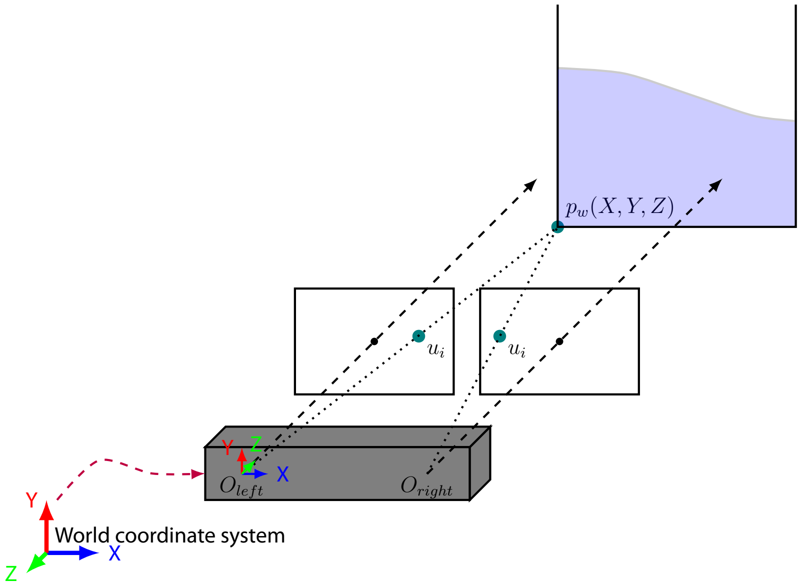


Figure 3.7: The picture shows the functioning of the stereo system in this work. We fix the location of the camera, which is referenced to the origin position through the extrinsic parameters. The two images provided by the stereo system output the desired 3D reconstruction of a point. The camera performs continuous triangulations and exports the depth of each pixel from the 2D matches detected between the right and left lenses.

3.7.1 Feature detection and tracking

We are interested in the detection and tracking of the sloshing of liquids. They are usually contained in glasses, vessels, and cups, that shape them and enable their manipulation. The movement of the container in the manipulation triggers the slosh of the fluid. Therefore, by tracking the container we can perform an estimation of the phenomenon under study.

However, transparent objects, such as those made of glass, have always entailed extreme difficulty for feature extraction algorithms due to their lack of texture. Only recently techniques based on deep convolutional neural networks, CNN, have been developed to overcome this problem [Khaing and Masayuki, 2018] [Sajjan et al., 2019]. Instead of using fiducial markers [Fujisawa and Kato, 2009], we decided to add texture to the glass to track the features or relevant points as seen in Fig. 3.1. Little dots were painted on the glass to create points of interest that the feature detector could select. From the detection of these features, or points of interest, we localize the center of masses of the glass projected to its bottom surface. We apply the Shi-Tomasi algorithm [Shi et al., 1994] for feature extraction in the area where the glass is expected to be. It finds the strongest, and more stable, features to track along the video sequence. The camera straightly provides the 3D position of the selected points of interest. With those points, we compute the center of masses of the glass projected to the bottom of the container. By tracking that point, we obtain information on the position and velocity of the glass.

The presentation of results coming from the cognitive digital twin may be useful for a potential user. In the use of simulation as the engine for physical scene understanding, we have access to predictions of the state of the dynamical system, including quantities of interest that are not accessible visually, such as the energy or the stresses, and that could be useful in critical situations for decision making.

3.7.2 Simulation in the loop and twin representation

The workflow of the twin is defined as a logical sequence that allows going from an input impulse to the physical container to the prediction of its effect in the virtual fluid. The resulting simulation will then be superimposed on the glass image. Firstly, we estimate the velocity of the glass based on feature extraction. The simulations of the database have been obtained by defining different input velocities of the glass that triggered the sloshing. Therefore, as a second step, we relate the initial state of the virtual fluid and the velocity of the container as an interpolation. This process converts the scene information into an input interpretable for the model.

Finally, the result is presented to the user in the scene using augmented reality. The augmented reality interface is an advantageous tool for result representation, as well as for model interaction. It provides a user-friendly interface for control and understanding. This technology is usually employed for rigid representation, i.e. the virtual object does not interact with any real stimulus. Our twin is deformable and interacts actively with the scene. We employ the tracking information previously obtained for the precise placing of the augmented liquid. In consequence, it continuously updates and shows real-time connection and interaction with the glass. AR representation shows the position of the particles and the free surface. In addition, it can augment the representation of the liquid by showing additional information, such as the velocity field simulated for the next time steps.

The simulation loop performs as follows. We detect the velocity of the glass and interpolate the initial state of the fluid. The current vector of state variables is projected to a low-dimensional manifold where the calculations are performed. We obtain the predicted new state of the fluid after 0.03 seconds of simulation in real-time. Then, the result is represented in the full order space and represented using AR. This process is repeated cyclically, with the help of AR for human-machine interaction. As a result, we have a real-time digital twin from the correlation between the movement of the glass and the state of the fluid.

3.7.3 Detection and data acquisition of the free surface

Some features of the algorithm also require data acquisition of the free surface. As part of the fluid recognition stage, we also need to acquire data from the fluid. Due to its lack of texture, or even transparency, obtaining information is problematic. The use of RGB Depth cameras is commonly used for this purpose [Do et al., 2016] [Tosun et al., 2017]. However, due to its lack of texture, it is a difficult task that could need to be supported with CNN for accurate tracking [Schenck and Fox, 2016a].

We do not consider adding texture to the liquid since this action is difficult to control, and would alter the material properties. Finally, we followed an approach similar to the one employed in [Eppel, 2016]. We analyze the color gradient of the binarized frame (Fig. 3.8 and 3.9). There is a noticeable change between the free surface and the background. Therefore, the points in the boundary where there is a color gradient are considered free surfaces. These points are stored, tracked, and augmented in the image for user interaction and verification.

3.8 Experimental validation of the twin

We have tested the online performance of the twin to evaluate the implementation of the algorithm in conjunction with the computer vision techniques that we employ for data extraction.

We obtained positive results in the merging of both elements. Trajectory generation and augmented representation is done online and coupled with the video. As a result, we perform real-time calculations and representation.

The replica of the liquid has been also compared with a glass filled with a liquid of the same type. Both containers are subjected to the same forces. Qualitatively, liquids are synchronized. Nevertheless, the movement of the digital twin seems to be a bit more amplified. While previous approaches in the field (notably [Schenck and Fox, 2016a] [Schenck and Fox, 2018a] [Kennedy et al., 2019]) report qualitative performance measurements only, we have also tried to provide quantifiable results to perform accurately the experimental validation. We quantify the error in the reproduction of the free surface reconstruction, defined as the integral of the differences between the heights regarding a middle line, see Fig. 3.10:

$$e = \frac{1}{l} \int_l \left(\left| h^R(x) - h^V(x) \right| \right) dx,$$

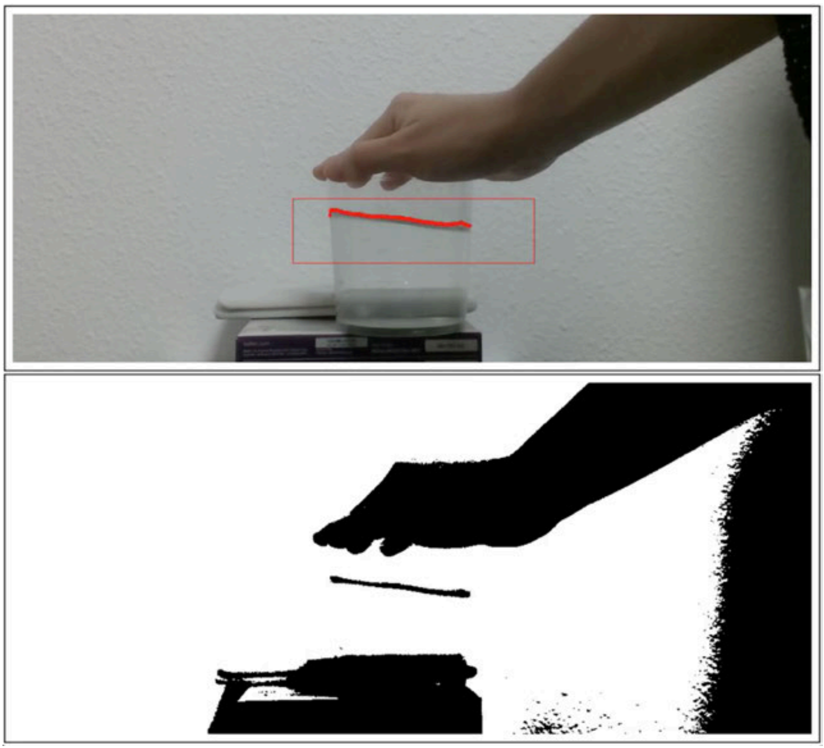


Figure 3.8: Example of frame binarization. The picture is first transformed to gray scale for gentle binarization. Noise is also filtered to detect a smooth surface.

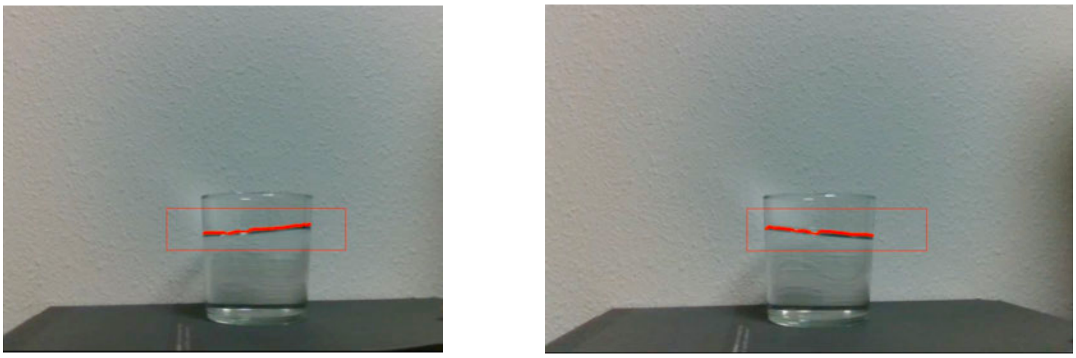


Figure 3.9: Free surface detection and tracking in video sequence. The points selected to belong to the free surface are highlighted in red over the original frame for verification.

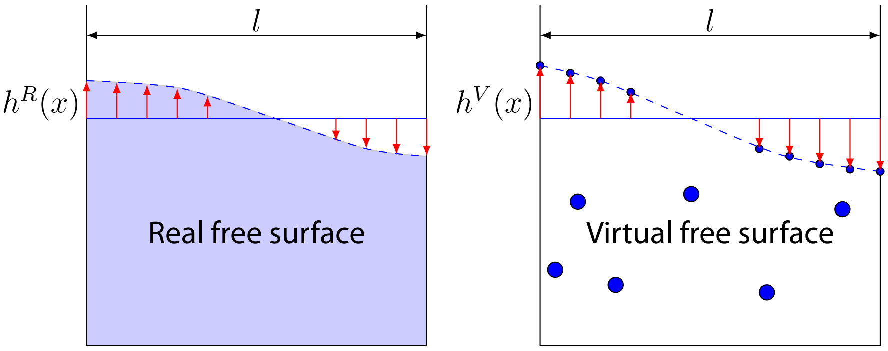


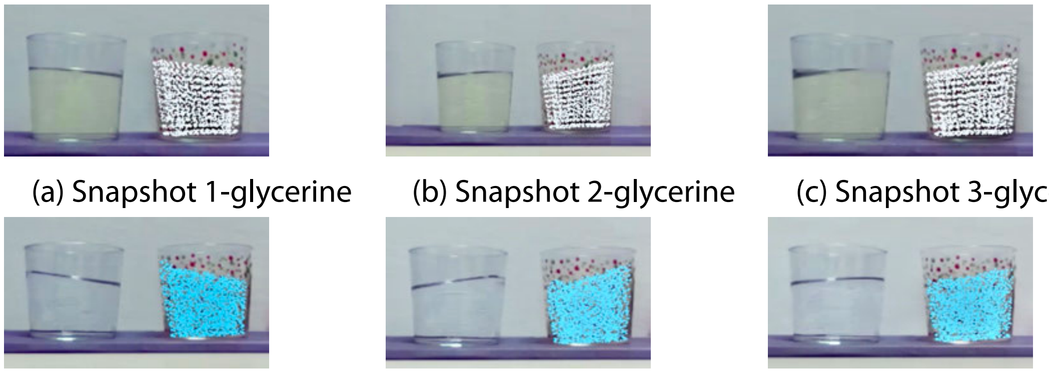
Figure 3.10: Representation of the quantifiable comparison of the real liquid and the replica. The free surface is defined as a function of its height at different points. These heights are compared in the same snapshot to evaluate the reconstruction error.

with R and V representing the real (physical) height and the virtual one, respectively.

The resulting errors are shown in Table 3.3. They remain adequate according to the state-of-the-art in computer vision applications. The error grows with higher amplitudes of slosh. Some sources of error could root in the approximation performed for velocity estimation. Remember that the pseudo experimental data with which we have built the model came also from a numerical approximation, SPH. Nevertheless, the resemblance is sufficient for learning a model, as well as corrections that will improve its performance.

3.9 Conclusions

In this chapter, we have presented and described a digital twin able to learn the sloshing dynamics occurring within a glass. We have shown that it satisfactorily reproduces the dynamics that a real stimulus would cause. The merge of online and real-time data acquisition, calculation, and result representation has enabled realistic interaction between the two mediums.



(a) Snapshot 1-glycerine (b) Snapshot 2-glycerine (c) Snapshot 3-glycerine

(d) Snapshot 1-water (e) Snapshot 2-water (f) Snapshot 3-water

Figure 3.11: Snapshots employed for comparison between the real liquid and the digital twin. The free surface reconstruction has been evaluated to compute the error.

WATER	$h^R(0)$	$h^V(0)$	$h^R(l)$	$h^V(l)$	mean error (mm)
Snapshot 1	5.52	11.32	5.38	8.28	0.7572
Snapshot 2	4.77	7.58	3.089	11.35	1.5254
Snapshot 3	3.694	2.839	2.839	6.53	1.0031
GLYCERINE	$h^R(0)$	$h^V(0)$	$h^R(l)$	$h^V(l)$	mean error (mm)
Snapshot1	3.56	3.845	3.56	3.418	0.119
Snapshot 2	4.20	5.95	3.18	8.85	1.7428
Snapshot 3	3.11	3.53	2.54	7.77	1.249

Table 3.3: Numerical results of the experimental validation. Absolute value measurements are provided at $h(0)$ and $h(l)$ in mm for comparison. The error is also expressed in mm.

The digital twin connects with the scene through computer vision techniques based on feature extraction to obtain the velocity of the container. This is the input of the simulations that we have performed to obtain synthetic data with which the model has been built. Therefore, we establish a straight relationship between scene data and interpretable inputs for the twin.

Real-time could not be achieved but by the use of model order reduction techniques. k-PCA finds a space of $3-4$ dimensions where the dynamics are embedded to perform the calculation with minimal loss of information. We have also proved the efficacy of GENERIC to learn more complex behaviors, such as viscoelasticity, widening the options that the twin offers.

While it has been difficult, in general, to obtain fully meaningful and quantitative comparisons with existing methods—that, in addition, focus on the pouring process, while we are interested in the sloshing phenomenon—, our method guarantees by construction the fulfillment of the laws of thermodynamics while bypassing the integration of Navier-

Stokes equations. This has been shown to provide very accurate results.

Given these tools and the results obtained from the classification training, new capabilities can be added to the twin. We expect to feed the algorithm with data from the free surface of a real fluid to work in limited and small data regimes. Also, the model could be corrected in case the liquid is unknown to go a step further and transform the model into a hybrid twin.

Chapter 4

On the use of thermodynamics as an inductive bias for neural networks

Physics perception very often faces the problem that only limited data or partial measurements on the scene are available. In this chapter, we propose a strategy based on neural networks to learn the full state of sloshing liquids from measurements of the free surface. Our approach is based on recurrent neural networks (RNN) that project the limited information available to a reduced-order manifold to not only reconstruct the unknown information but also be capable of performing fluid reasoning about future scenarios in real-time. To obtain physically consistent predictions, we train deep neural networks on the reduced-order manifold that, through the employ of inductive biases, ensure the fulfillment of the principles of thermodynamics. RNNs learn from history the required hidden information to correlate the limited information with the latent space where the simulation occurs. Finally, a decoder returns data to the high-dimensional manifold, to provide the user with insightful information in the form of augmented reality.

This work has been published in the next contribution:

- Moya, B., Badías, A., González, D., Chinesta, F., & Cueto, E. (2021). Physics perception in sloshing scenes with guaranteed thermodynamic consistency. arXiv preprint arXiv:2106.13301. Accepted for publication in IEEE Transactions in Pattern Analysis and Machine Intelligence.

4.1 Introduction

In the context of physics perception, world simulation is the emulation of the real world in a virtual environment for machine understanding. Simulations provide an interpretation of data inputs and provide an estimation of the consequences of actions. However, we do not always have access to the whole set of state variables that the employed description may require. In this next step, we aim to connect with the real entity of a liquid to interpret the sloshing behavior directly from the information of the fluid, and not that of the container. As a result, the simulation engine must be able to work with the data acquired from the stereo camera, which consists in measurements of the free surface.

Hence, the main constraint in this step is the availability of data. Techniques in modeling fluid reasoning continue to expand, but there are access limitations to full sets of experimental variables required in the proposed descriptions. That is the case with our methodology. It operates from full-field descriptions of the dynamical state of the fluid, i.e. evaluation of the state variables at each particle of the discretization of the fluid. We do not consider sophisticated tools, such as PIV cameras, to make our approach applicable to different scenarios. We solely work from the measurements available from our stereo camera. Thus, we only have access to evaluations of the free surface in time.

This is not an isolated problem in fluid dynamics. Data-driven modeling is sometimes conditioned by the scarcity of these types of measurements. As a solution, some works propose to build models upon the information available in images [Bai et al., 2017] [Rodríguez-Ocampo et al., 2020] or sensors [Bieker et al., 2020]. For instance, [Flaschel et al., 2021] developed an algorithm for physically-consistent model inference with information about displacements and force. However, in some cases, this approach could jeopardize the physical consistency of the method if the dynamical state is not reconstructed. In contrast, we propose an image-based method to acquire data with computer vision techniques and recover the dynamical features of the slosh to preserve the soundness of the algorithm.

From this perspective, the strategy proposed could be seen as a self-supervised learning technique to overcome the lack of labeled data. Labeled data is a bottleneck for data-driven model inference. Despite this, we can look for methods that achieve a deeper understanding of the data available to recover the required information. This problem is of utmost importance in the field of scene perception and understanding [Schenck and Fox, 2018b] [Nair et al., 2017] [Nava et al., 2021] [Yan et al., 2020]. An appealing option is the development of strategies that recover the dynamic information from the data acquired [Rao et al., 2021]. Sensors are usually placed strategically to recover the solution field of interest [Callaham et al., 2019] [Sun and Wang, 2020]. Deep neural networks are widely used for this task. [Erichson et al., 2020] suggests the use of shallow neural networks for reconstructing fluid flows. In a different approach, [Lye et al., 2020] perform an estimation of the input parameters of fluid flows in a turbulent regime from measurable information.

In our approach, the algorithm is fully phrased with deep neural networks for this purpose. We find these techniques convenient not only for modeling but also for recovering the dynamical features from partial measurements. We first build a low-dimensional manifold based on autoencoders. These are unsupervised learning techniques that achieve compression of the information by learning a low-dimensional manifold of the data provided. Then we model the dynamics in the framework provided by GENERIC using the so-called Structure-Preserving Neural Networks (SPNNs) [Hernández et al., 2021b] [Hernández et al., 2021a]. This implementation consists in a perceptron constrained by the degeneracy conditions and the learning scheme provided as the inductive bias of the network. Finally, we hypothesize that one single snapshot of the evaluation of the free surface may not contain enough information to recover unveil the internal features of the dynamics. However, the history of the free surface underlies the influence of the hidden variables whose information we have to access. Recurrent neural networks are structures specialized in distilling information in sequences of data. Hence, we aim to find a correlation with this type of network to find a mapping between sequences of the free surface of the liquid and the low dimensional manifold learned from full-field evaluations of the state variables.

Recent works have also learned dynamical descriptions inspired by the GENERIC formalism [Lee et al., 2021] [Zhang et al., 2021]. The main difference with SPNNs is the imposition of hard constraints for learning the degeneracy restrictions, one of the prerequisites in this formulation to have an adequate structure. In contrast, the proposed method opts for the optimization of this criterium as a soft constraint that facilitates optimization and convergence.

There exist other formulations based on thermodynamical priors [Vlassis and Sun, 2021]. Yu et al. [Yu et al., 2021] developed a learning scheme based on the Generalized Osanger principle to describe the evolution of the system in terms of energy and entropy potentials. The Generalized Osanger Principle is adapted to data-driven inference with a Runge-Kutta scheme. Thermodynamically informed neural networks (TINNs) [Masi et al., 2021] [Masi et al., 2020] define by automatic differentiation the derivatives of the network solution with respect to the outputs to evaluate the fulfillment of the laws of thermodynamics.

Although the algorithm is initially trained with computational data, the final goal is to connect it with real liquids to close the perception loop. Detection and tracking of fluids, as well as containers, may be difficult if they lack texture. The measurements obtained are usually invalid or noisy because the surfaces are not Lambertian [Koppal, 2014]. We are interested in the detection of fluids, particularly the free surface. [Schenck and Fox, 2016a] and [Schenck and Fox, 2018a] propose the use of CNNs to perform tracking of the fluids. In the work of [Do et al., 2016], the authors propose an algorithm for filling level detection with RGD-D cameras. We propose an approach similar to the one presented in [Eppel, 2016]. We convert the color image into a binary image in black and white to detect the color gradient that appears on the free surface.

4.2 Problem description

In this chapter, we introduce a learned simulator that, in contrast with similar recent strategies, see [Wu et al., 2015] for instance, outputs augmented information that can quantitatively improve reality for decision making given partial measurements. The starting point is similar to the one stated in previous chapters. We perform computational simulations under different initial velocities to trigger the slosh to prepare the dataset. Based on the GENERIC formalism, we describe the fluid dynamics in terms of a set of state variables to fully describe the thermodynamical state of the liquid. Therefore, we choose the 3D position, velocity, internal energy, and stress tensor of each particle at each time step of the simulation to conform to the dataset.

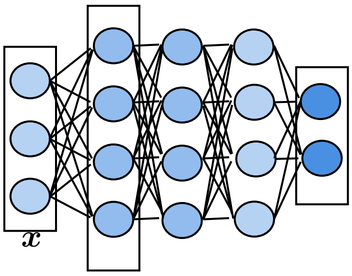
Then, we propose a three-step strategy to build an algorithm that, coupled with the real liquid, integrates the evolution in time of the slosh given measurements of the free surface only. Particularly, we have access to the pixels of the free surface that we can project to 3D real-world coordinates. First, we learn a reduced-order representation of the dynamics from the computational dataset by employing autoencoders to compress the information of the initial dataset from full field state vectors, i.e. considering observable and nonobservable data. Secondly, we train the time integrator of the evolution of the dynamics in the latent manifold based on the results just obtained. Finally, the encoder part of the autoencoder is substituted by a recurrent neural network (GRU specifically) to distill from the time evolution of the free surface the dynamical insights necessary to project the current state of the fluid to the latent manifold from limited data. Thus, the flow of information consists in detecting the free surface of the fluid, assembling a sequence of snapshots, projecting the sequence to the low dimensional manifold, integrating the dynamics in time, and projecting the simulation results to the high order space. The training of the RNN is performed a posteriori, after the autoencoder has been trained, inferring the knowledge acquired about the latent space.

The proof-of-concept proposed is a glass full of glycerine. It is worth noting that the geometry of both the glass and the liquid are not a parameter of the problem and we opt for the use use of fully connected neural networks instead of graph-inspired architectures. Provided that we always have the same liquid volume, we always have the same discretization. In addition, despite the evolution of the particles in the liquid simulations based on SPH, their dynamics for the proposed fluid are not extremely chaotic. However, the liquid particles, and especially the free surface, do experience variations that have to be learned by the autoencoder and the physical model. In this stage, we are focusing on the implementation of the physics algorithm and the limitations of data of the dynamics. The snapshots of the sloshing solutions are assembled by assigning a tag to each particle to propose a specific order for the state vector. The particles are referred to as a local coordinate system located at the bottom of the cup. The final goal is to couple this algorithm with the liquid and to perform real-time simulations of new, previously unseen, fluids.

4.3 Artificial Neural Network Theory

Artificial Neural Networks (ANNs) are widely employed in diverse areas such as speech recognition, image classification, robotics, and quantum physics. However, they are a

$$f(\boldsymbol{z}) = f_n \circ f_{n-1} \circ \ldots \circ f_1(\boldsymbol{z})$$



$$f_i(x) = \text{act}(\omega_i \boldsymbol{z}_{i-1} + b_i)$$

Activation function act

$\text{ReLu}(x) = \max\{0, x\}$
$\text{Tanh}(x) = \ e^z - e^{-z} / e^z + e^{-z}$
$\sigma(x) \ = \ 1/\ 1 + e^{-z}$

Figure 4.1: Graphic representation of the mathematical elements in ANNs.

source of intense development and investigation due to their effectiveness at learning predictions in dynamical approximation problems, and their use has been popularized in recent years [Su and Yang, 2002] [Ghaboussi, 2010]. The progressively increasing employment of these methods is partially due to the *Universal Approximation Theorem*, which states the generality and universality of neural networks to approximate any function at a prescribed level of accuracy [Hornik et al., 1989]. As it is an approximation, the solution proposed may not be exact, but we can improve the solution by increasing the number of layers and neurons to fit the problem. The n-layer neural network proposed for each problem represents a mathematical function f, which results from the combination of multivariate functions $f_1, f_2, ..., f_n$ and an output function g. f is a mapping $f : \mathbb{R}^{D_\text{in}} \to \mathbb{R}^{D_\text{out}}$, with D_in and D_out the dimensionality of the input and the output respectively. f, for n layers, is defined as:

$$f = f_n \circ f_{n-1} \circ \ldots \circ f_1,$$

where each f_i is:

$$f_i : \mathbb{R}^D_{i-1} \to \mathbb{R}^D_i,$$
$$f_i = \text{act}(w_i \boldsymbol{z} + b_i),$$

being w_i and b_i the weights and biases, i.e. the parameters θ subjected to the optimization, and act, being the activation function that defines the nonlinearity. This scheme has been displayed in Fig. 4.1 for the sake of clarity.

4.4 Method

The complexity of the algorithm just presented forces us to implement our system in three different steps, see Fig. 4.2 for a graphical sketch of the implemented architecture. The highly dimensional nature of the problem motivates the reduction of the dynamics to carry out learning on an embedded space of a much lower dimension. In the case of

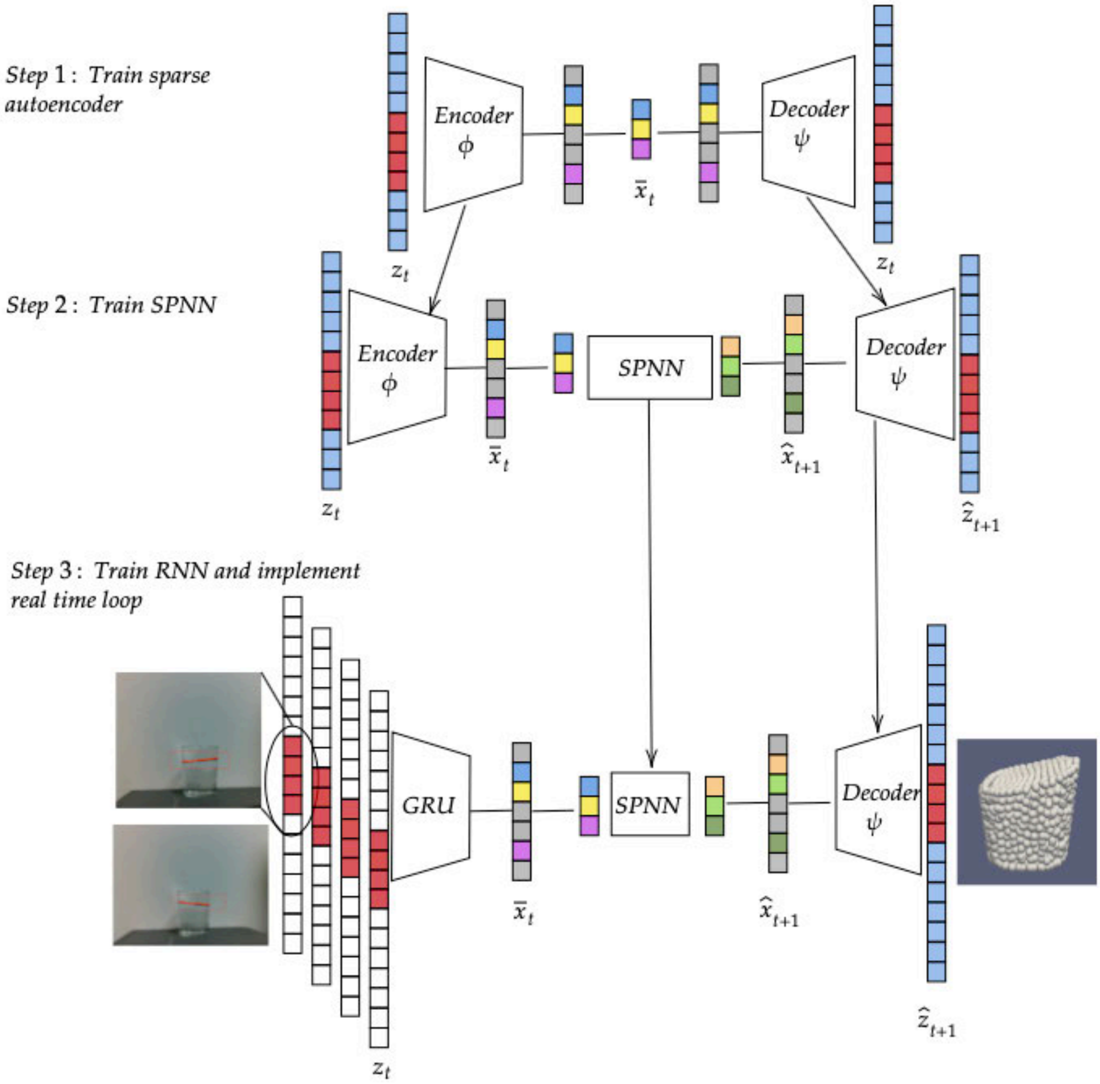


Figure 4.2: Sketch of the construction procedure for the deep neural network. In the first step (first row), we perform model order reduction with autoencoders. Then, we train a structure-preserving neural network to integrate in time the state of the system. We do not have access to the high-dimensional state of the system, only a portion of it, in red in the input vector. The encoder is substituted by a RNN to find a mapping from the partial measurements to the low-dimensional manifold.

learning and predicting new situations from real-world data, a correlation needs to be established between the available data (the free surface) and the latent space built from full computational descriptions. We hypothesize the existence of features (distinctive attributes) in data sequences of the free surface that relates the history of partial measurements with internal variables of the fluid. An architecture based on recurrent neural networks can unveil those correlations, map the data acquired to the latent space, and output the reconstructed state in the next time step.

To accomplish these requirements, we need to develop three different architectures. Firstly, we have to project our computational data to a lower-dimensional manifold to train the algorithm efficiently and achieve real-time performance in the implementation. Secondly, we need to train a physics-informed integrator that will learn the evolution of the dynamics in the latent space. Finally, we work over an algorithm based on recurrent neural networks to substitute the encoder with a perceptron capable of projecting partial measurements to the latent space. It is worth mentioning that we chose to train in two different steps the autoencoder and the RNN and apply transfer learning. In transfer learning, we profit from the knowledge acquired in a new application. Sloshing dynamics is already a complex problem to apply model order reduction, and working with limited information from the free surface could complicate the process of learning a low-dimensional manifold. Despite these considerations to optimize the algorithm, the need for deep neural networks that learn the patterns of the dynamics remains.

4.4.1 Model order reduction with autoencoders

Despite the success of neural networks in modeling fluid dynamics, the convergence and accuracy of the training could be put at risk due to the high dimensionality of the description of the discretized fluid. In addition, fluid dynamics are led by strong non-linear structures hard to be learned by machine learning methods. Autoencoders are a specific type of neural network architecture to apply model order reduction. As a method of unsupervised learning, it learns a compression of the information provided to describe the information in a latent manifold of much lower dimensionality than the original database. The advantage of its use compared to other techniques rests, firstly, with the facility to connect the different parts of the algorithm as a *metamodel*, understood as an algorithm that works from inputs of different models connected. In addition, the levels of accuracy are comparable to other non-linear model order reduction techniques. Autoencoders capture the nonlinear pattern that emerges in the database due to the deep of the network and the nonlinear activations of the layers.

Autoencoders consists of two parts: an encoder that maps data to an embedded space, and a decoder that reconstructs latent information to the original space. Fig. 4.3 shows a graphical representation of an autoencoder. The output of the decoder $\hat{z}_t = \hat{z}(t)$ has to

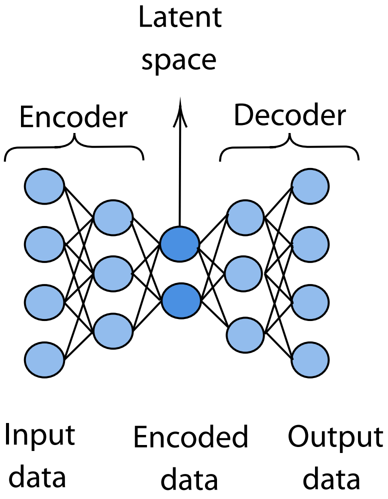


Figure 4.3: Graphical description of an autoencoder. The encoder learns a mapping to a latent space where the information of the input is compressed to a low dimensional representation. The decoder, usually symmetric to the encoder, undoes the process to map the hidden state to the full space.

be equal to the input of the encoder $\boldsymbol{z}_t = \boldsymbol{z}(t)$,

$$\begin{aligned} \text{Encoder } \phi : \mathcal{M} \subset \mathbb{R}^D &\to \mathbb{R}^d \\ \boldsymbol{z} &\mapsto \boldsymbol{x}, \end{aligned}$$

$$\begin{aligned} \text{Decoder } \psi : \mathcal{N} \subset \mathbb{R}^d &\to \mathbb{R}^D \\ \boldsymbol{x} &\mapsto \hat{\boldsymbol{z}}. \end{aligned}$$

We decided to specifically employ sparse autoencoders (SAE). They include a L_1-norm penalization so we enforce sparsity in the bottleneck to look for the intrinsic low dimensionality of the latent manifold [Ng et al., 2011]. From the perspective of scientific machine learning, this can be seen as a way of imposing parsimony—in other words, the simplest and sufficient representation of the model, like Occam's razor—to the learned model [Liu and Tegmark, 2020].

We train the model through the backpropagation of the combination of two losses. Given N_{snap} snapshots introduced in the algorithm, the first loss term $\mathcal{L}^{\text{sae}}_{\text{mse}}$ refers to the reconstruction error between the ground truth and the result of the decoder, evaluated with the mean squared error (MSE)

$$\mathcal{L}^{\text{sae}}_{\text{MSE}} = \frac{1}{N_{\text{snap}}} \sum_{i=1}^{N_{\text{snap}}} (\boldsymbol{z}_i - \hat{\boldsymbol{z}}_i)^2. \tag{4.1}$$

The second term of the loss is a regularizer term $\mathcal{L}^{\text{sae}}_{\text{reg}}$ to impose the sparsity in the bottleneck $\boldsymbol{x}_i$,

$$\mathcal{L}^{\text{sae}}_{\text{reg}} = \sum_{i=1}^{N_d} |\boldsymbol{x}_i|, \tag{4.2}$$

where N_d is the dimension of the low dimensional manifold learned by the autoencoder. The size of the bottleneck is fixed a priori, and the number of non-vanishing entries (i.e., the intrinsic dimensionality of data) will be determined without user intervention during the training period.

The contribution of the regularization is weighted with a coefficient $\lambda^{\text{sae}}_{\text{reg}}$ to control its influence in the training process,

$$\mathcal{L}^{\text{sae}} = \mathcal{L}^{\text{sae}}_{\text{MSE}} + \lambda^{\text{sae}}_{\text{reg}} \mathcal{L}^{\text{sae}}_{\text{reg}}. \tag{4.3}$$

Previous to this step, we normalized the dataset through escalation to have values in the range of $(-1, 1)$. However, we do not build a global autoencoder for the whole dataset, but five individual SAEs for each group of state variables (position, velocity, internal energy, and stress tensor separated in normal $\boldsymbol{\sigma}$ and shear $\boldsymbol{\tau}$ components). The goal is to capture the features of each group with an independent autoencoder. The final bottleneck consists of the merge of the individual latent manifolds learned.

4.4.2 Structure-preserving neural networks

In previous chapters, GENERIC was learned as part of an optimization problem to infer the constitutive manifold of the sloshing to interpolate the dynamical evolution embedded in this manifold. However, deep neural networks learn the GENERIC evolution of the system from a different perspective. They learn a nonlinear mapping to the aforementioned constitutive manifold to output the constituents of the formulation to perform the integration in time.

As in the previous approach, we start from the discretization of the employed formalism and the degeneracy conditions to infer them from data

$$\frac{\boldsymbol{x}_{n+1} - \boldsymbol{x}_n}{\Delta t} = \mathsf{L}_n \mathsf{DE}_n + \mathsf{M}_n \mathsf{DS}_n, \tag{4.4}$$

where L, M, DE and DS represent the discretized versions of $\boldsymbol{L}$, $\boldsymbol{M}$, ∇E and ∇S, respectively, $\boldsymbol{x}$ represent the state vector in the low-dimensional manifold, and the subscript n refers to time $t = n\Delta t$ and $n+1$ indicates time $t + \Delta t$, respectively.

Structure-preserving neural networks (SPNN) embed GENERIC into a deep neural network to learn the value of the discretized gradients of energy and entropy, which are the targets of the optimization in the present deep learning approach. Since we learn the constitutive manifold in a reduced representation of the dynamics, L and M are also part of the optimization. Thus, the proposed architecture unveils L, M, DE and DS from data. The SPNN consists of fully-connected layers in a feed-forward flow that learns a mapping to the

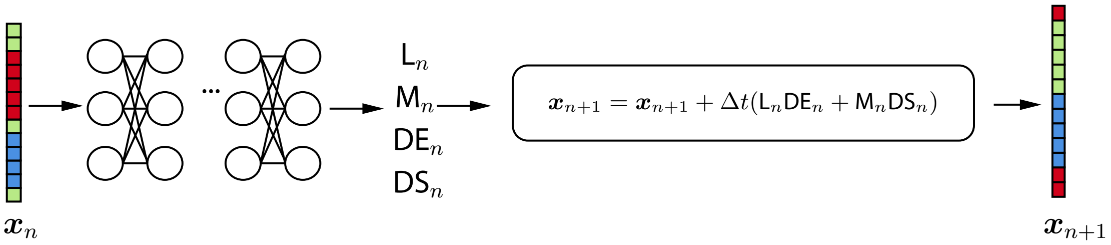

L_n
M_n
DE_n
DS_n
$\boldsymbol{x}_{n+1} = \boldsymbol{x}_{n+1} + \Delta t(\mathsf{L}_n\mathsf{DE}_n + \mathsf{M}_n\mathsf{DS}_n)$
$\boldsymbol{x}_n$
$\boldsymbol{x}_{n+1}$

Figure 4.4: Scheme of structure-preserving neural networks (SPNN). We introduce the vector of state variables $\boldsymbol{x}_n$, in the reduced space, into the network to learn $\mathsf{L}_n, \mathsf{M}_n, \mathsf{DE}_n, \mathsf{DS}_n$ and perform the time integration. The final output is the vector of state variables $\boldsymbol{x}_{n+1}$, in the reduced space.

quantities of interest. This learning scheme is constrained by the degeneracy conditions and the imposition of skew-symmetry and symmetry of L and M respectively.

We provide pairs of snapshots in t and $t + \Delta t$ to learn the integrator of the sloshing dynamics. The input of the neural network is the input state vector in the reduced-order manifold $\boldsymbol{x}_n$. The net outputs a solution vector that contains the L, M, and gradients of energy and entropy associated to $\boldsymbol{x}_n$. Then, we have the elements to integrate the dynamics in time. The flow of information following is presented in Fig. 4.4.

We guide the training with two losses that we backpropagate through the networks. We first evaluate the performance of the time integration given the predicted gradients and matrices comparing the ground truth $\boldsymbol{x}_{n+1}$ with the solution $\hat{\boldsymbol{x}}_{n+1}$ with the L_2-norm. This loss is weighted with the hyperparameter $\lambda^{\mathsf{SPNN}}_{\mathsf{MSE}}$ to control its influence in the global loss function of the network. The second loss term penalizes the deviation from the degeneracy conditions. This is evaluated as a sum of the squared values of the two contributions, related to the energy conservation and entropy production

$$\mathcal{L}^{\mathsf{SPNN}}_{\mathsf{deg}} = \frac{1}{N_{\mathsf{snap}}} \sum_{i=1}^{N_{\mathsf{snap}}} (\mathsf{L}_i\mathsf{DS}_i)^2 + (\mathsf{M}_i\mathsf{DE}_i)^2. \tag{4.5}$$

4.4.3 Recurrent neural networks to recover dynamical information

A challenge in computer vision is the impossibility to evaluate dynamical internal variables required in physics-informed learning. In the present approach, the latent manifold of the dynamics is trained from full-field data so that we applied the learning algorithm in a thermodynamically admissible context for the proposed method. Even considering complementary devices to support the acquired data, we still do not have direct access to essential magnitudes in the description, such as the internal energy.

We work from a self-supervised learning perspective to unveil the information not provided about the dynamical state of the fluid. We propose to extract the dynamic features that are involved in the evolution of the free surface from the study of its history. For this purpose, we suggest the use of a recurrent neural network that correlates the reduced-order manifold of the dynamics with the partial measurements of the free surface. Recurrent neural networks learn from the sequences of information instead of individual snapshots. These architectures are commonly present in works of natural language processing, speech recognition, or economics. Conversely, vanilla recurrent neural applied to more complex problems usually experience vanishing and exploding gradients [Pascanu et al., 2013]. As a solution to this problem, the community proposes two next-generation recurrent neural networks: Gated Recurrent Units (GRU) [Cho et al., 2014] and Long Short-Term Memory (LSTM) units [Hochreiter and Schmidhuber, 1997]. They include additional flows of information, known as *gates*, that preserve long-term dependencies and avoid learning short-term features.

GRUs are similar to LSTMs networks, but they lack the *forget gate* that the latter has. GRUs are thus simpler structures that involve fewer parameters. However, the performance of both is comparable in certain cases. In addition, GRUs have proven to train faster and more efficiently with smaller datasets and shorter sequences [Chung et al., 2014].

The basic idea behind the GRU architecture is to accumulate information from previous layers, see Fig. 4.5. The hidden state $\boldsymbol{h}_t$ represents a summary of the features identified in previous sequences. $\boldsymbol{g}_t^{\text{update}}$ is the output of the update gate. This gate selects which information from the hidden state and the input sequence passes to the next step, modeled with a sigmoid activation function. In contrast, the reset gate $\boldsymbol{r}_t$ reflects the past information that should be avoided. A new memory cell $\boldsymbol{n}_t$, defined as the reset information, stores only the relevant information from the past. These layers are defined by the parameters U and W. The output is the final hidden state $\boldsymbol{h}_t$ that accumulates the relevant information of past states and features learned from the current input sequence:

$$\begin{aligned}
\boldsymbol{g}_t^{\text{update}} &= \sigma(\boldsymbol{x}_t U^z + \boldsymbol{h}_{t-1} W^z), \\
\boldsymbol{r}_t &= \sigma(\boldsymbol{x}_t U^r + \boldsymbol{h}_{t-1} W^r), \\
\boldsymbol{n}_t &= \tanh(\boldsymbol{x}_t U^h + (\boldsymbol{r}_t \boldsymbol{h}_{t-1}) W^h), \\
\boldsymbol{h}_t &= (1 - \boldsymbol{g}_t^{\text{update}}) \boldsymbol{h}_{t-1} + \boldsymbol{g}_t^{\text{update}} \boldsymbol{n}_t.
\end{aligned}$$

The RNN input is a sequence of the vertical and horizontal positions of selected, equally spaced, points of the free surface. We perform the training of the mapping with the computational data available. We track the particles that belong to the free surface with an algorithm that compares the height of the particles and their neighbors. Then, we interpolate the vertical displacement of selected points of the profile to compare the free surface of different dynamical states with a uniform mesh. We select 21 points of the free surface and assemble sequences of these snapshots. The batch of sequences is introduced in the network to pass through GRU recurrent layers doing a projection *from-many-to-one*, i.e., introducing a sequence to obtain a single vector as output, that corresponds to the latent space. The output vector of the GRU layers passes through a final forward fully connected layer with linear activation. The result of this process $\hat{\boldsymbol{x}}_t$ must match the latent state vector

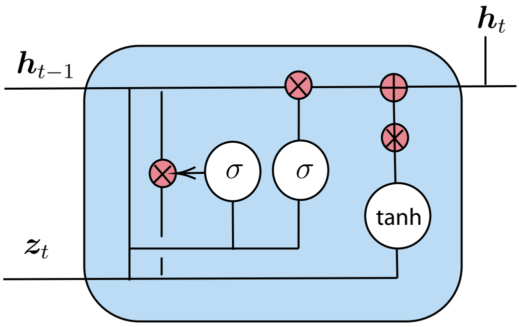


Figure 4.5: Representation of a GRU cell. The three main paths indicated represent the update and reset gates, the new memory cell, and their connection to update the new hidden state transmitted to the next later.

corresponding to the last snapshot $\boldsymbol{x}_t$ of the given sequence. The loss $\mathcal{L}^{\text{GRU}}_{\text{mse}}$ evaluates the MSE between the predicted latent state and the ground truth,

$$\mathcal{L}^{\text{GRU}}_{\text{MSE}} = \frac{1}{N_{\text{snap}}} \sum_{i=1}^{N_{\text{snap}}} (\boldsymbol{x}_i - \hat{\boldsymbol{x}}_i)^2. \tag{4.6}$$

4.5 Computational training and validation

We split the computational dataset of glycerine, composed of 1600 snapshots, into two subsets: 80% of the snapshots are employed for training and the remaining 20% for testing. As a reminder, we have four sloshing simulations of 2 seconds discretized in time increments set to 0.005 seconds. This time increment is the one defined for the integration algorithm. We use the same train and test datasets for the three networks developed.

The results from one training are transferred to the next training. Finally, the networks are assembled to build the simulation loop of the dynamics from the partial measurements of the liquid, to the output of the next state of the fluid including the reconstructed fluid and the velocities, stress, and energy fields.

4.5.1 Hyperparameters and training details

Each input vector includes the position, velocity, internal energy, and stresses (shear and normal) evaluated of each particle of the SPH discretization. Each particle has a label, and the state vector is assembled according to this order. The fluid consists of 2134 particles. As a result, the full dimensionality of the state vector is 27742.

Each of the five SAEs (one for each variable group) is initialized following the Kaiming method. In this approach, the weight initialization follows a Gaussian, and biases of the network are set to zero [He et al., 2015]. Encoder and decoder have symmetric structures,

	lr	wd	λ^{sae}
Position ($\boldsymbol{q}$)	10^{-4}	10^{-6}	10^{-3}
Velocity($\boldsymbol{v}$)	10^{-4}	10^{-5}	10^{-3}
Internal energy (e)	10^{-4}	10^{-5}	10^{-4}
Normal stress ($\boldsymbol{\sigma}$)	10^{-4}	10^{-5}	$5 \cdot 10^{-3}$
Shear stress ($\boldsymbol{\tau}$)	10^{-3}	10^{-6}	$5 \cdot 10^{-3}$

Table 4.1: Training parameters for each SAE.

and we apply in both ReLU activation for the hidden layers, a linear activation of the input and output layers. We use Adam optimizer, and update the learning rate at 1000 and 3000 epochs of training. The hyperparameters (learning rate lr, number and size of layers, weight decay wd and sparse weights λ^{sae}) are defined independently for each group of state variables provided its complexity and proper features. Learning rates and weight decays selected are displayed in Table 4.1. We set low learning rates due to the complexity of the patterns of the state variables:

- Position: Input size is $D = 6402$ and output size $d = 20$. It is composed by $N_h = 2$ hidden layers of size 120.
- Velocity: Given the complexity of the velocity, we built a net of input size $D = 6402$, output size $d = 20$, $N_h = 4$ hidden layers, and hidden size 200.
- Internal energy: In the case of energy, input size is $D = 2134$, output size $d = 10$, and there are $N_h = 3$ hidden layers which consist of 40 neurons each.
- Normal stress: The normal stress tensor components are identical. Thus, the input shape of the net is $D = 2134$, the output shape is $d = 20$, and it is composed of $N_h = 3$ hidden layers of 200 neurons.
- Shear stress: This net had input size $D = 6402$, $N_h = 3$ hidden layers of 200 neurons, and output size $d = 20$.

We train for 10000 epochs. At this point, all the autoencoders converge to an optimal result. The dimensionality of each bottleneck is truncated by the sparsity imposed. The final dimensionality of the latent spaces of each autoencoder is: $d_{\text{position}} = 3$, $d_{\text{velocity}} = 3$, $d_{\text{energy}} = 2$, $d_{\sigma} = 3$ and $d_{\tau} = 2$. We merge the bottleneck of each reduced-order manifold, which results in a global manifold of $d_{\text{latent space}} = 13$ dimensions. This result is the input of the learned simulator and the output of the recurrent neural network in the meta-learning approach proposed. From this result, the SPNN and the RNN are trained simultaneously.

The SPNN is defined to output a vector including information of L and M and gradients of energy and entropy corresponding to the current latent vector of the dynamics. Providing that L and M are skew-symmetric and symmetric, respectively, we only learn the upper elements of the main diagonal. Thus, instead of learning the full matrices of dimension $d \times d$, we learn $d \cdot (d-1)/2$ elements for L and $d \cdot (d+1)/2$ elements for M. Therefore, considering that the gradients have size d, the final output size is $d_{out} = d \cdot (d-1)/2 + d \cdot (d+1)/2 + d + d = 195$. Working in a low-dimensional space is expected to favor the convergence of the network. However, the dynamical evolution is still complex. Consequently, the SPNN consists of $N_h = 13$ hidden layers, whose size is set to 195. Kaiming is the initialization method of the network, and we apply linear activation and ReLU activations to the output and hidden layers respectively. We define the learning rate and weight decay as $lr = 10^{-3}$, $wd = 10^{-5}$. We employ Adam optimizer for training the network, and the learning rate is updated at epochs 1500, 2400, and 4000. The reconstruction MSE loss is weighted by a factor $\lambda_{\text{mse}}^{spnn} = 10^3$ to prioritize the accuracy of the reconstruction in the backpropagation. The train finishes after 5000 epochs. At this point training and test losses are $3.2 \cdot 10^{-3}$ and $1.42 \cdot 10^{-2}$, respectively.

Although the time step of the discretization of the simulation is 0.005 seconds, the camera acquires data at 60 Hz, which is approximately 0.015 seconds. We decide to assemble the sequences and train the GRU according to this restriction for ease of coupling the algorithm with the real replica. The GRU consists of three recurrent layers of this type, of 26 neurons with ReLU activation, and a final fully connected layer, with linear activation, to accomplish the many-to-one scheme. We employ Adam optimizer and learning rate and weight decay are set to $lr = 10^{-3}$, $wd = 10^{-5}$. The learning rate is updated by a scheduler at 1000 and 3000 epochs. After testing with different sequences lengths, 16 was found to be the minimum number of snapshots required to find a mapping between the evaluations of the free surface and the low dimensional space.

4.5.2 Computational validation

The network is assembled after training and validated by providing a single input sequence from the dataset of glass velocity $v = 0.2$ m/s to trigger the slosh. The information is mapped to the low dimensional manifold to perform the time integration of the dynamics and output the reconstruction of the fluid and quantities of interest until we reach the steady state of the simulation. We could provide a sequence of data at each new time of the discretization. However, we test the stability and capacity of continuing the time integration if no more information is provided (due to occlusion, connection problems, ... for instance) by considering only that first snapshot.

Table 4.2 shows the MSE of the autoencoder proposed to reconstruct each group of state variables. These results have been compared with those obtained with POD [Ly and Tran, 2001] taking 10 modes, and k-PCA with 4 modes [Schölkopf et al., 1998]. Modes are selected concerning the evolution of the eigenvalues obtained from each method. The AE achieves the same or improved levels of accuracy as POD and k-PCA. Fig. 4.6 plots the simulation results in the reduced-order space. The initial state has been projected to the latent manifold to emulate the evolution of its behavior. Fig. 4.7 plots the time evolution

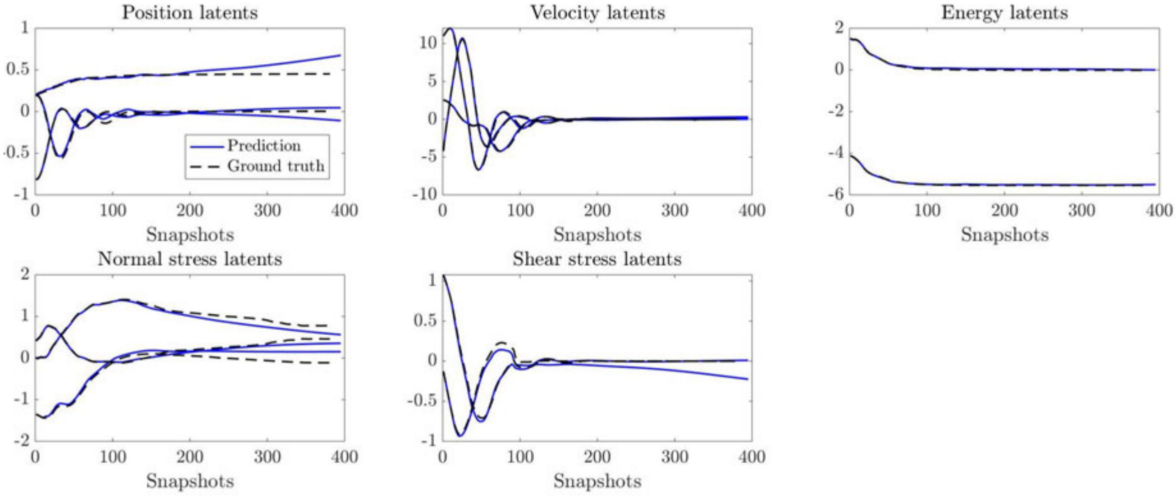


Figure 4.6: Simulation results. Learning of the dynamics in the latent manifold. Dashed lines represent the time evolution of the latents that aimed to be emulated. Lines in blue represent the result of the SPNN in the latent manifold.

of some state variables in the high-dimensional space for 21 randomly selected particles. Finally, Fig. 4.8 shows the comparison between the ground truth and the projection of the results to the high order manifold in three steps. This figure also includes the RMSE error of the reconstruction of each snapshot and the Hausdorff distance between the ground truth and the result. The Hausdorff distance (HD) evaluates the closeness of two sets by analyzing the largest distance between one set of points to another [Huttenlocher et al., 1993]. If the HD is low, it resembles a high degree of similarity

$$\mathrm{HD} = \max\left\{\sup_{x\in X} d(x,Y), \sup_{y\in Y} d(y,X)\right\},$$

being X and Y the two sets to be compared. In this context, the HD evaluates how well the shape of the result matches the shape of ground truth to have an indicator of the accuracy in the reconstruction. Thus, it compares the maximum (sup) distance from the ground truth to the output $\sup_{x\in X} d(x,Y)$, and vice versa $\sup_{y\in Y} d(y,X)$. Of these two distances, the maximum is the HD of the mismatch. After analyzing the results obtained in the computational phase, we decide to test the loop in a real scenario for the reconstruction of real fluids.

4.6 Coupling with the real liquid twin

Given the results obtained from the computational validation, the simulation loop is implemented to be coupled with the real entities of the liquids. The data acquired by the RGB-D camera is assembled into sequences to initiate the simulation of the time evolution of the liquid. As a result, we output the reconstruction of the fluid and the velocity,

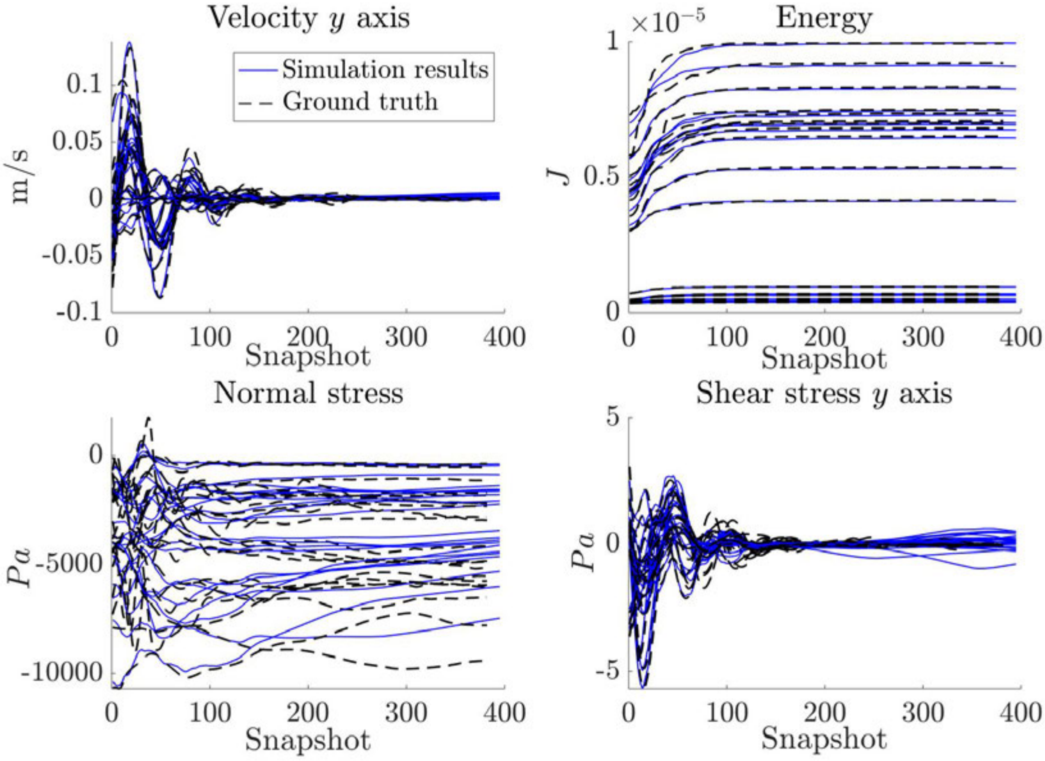


Figure 4.7: Time evolution of selected state variables evaluated at 21 random particles. The graph shows a comparison between the simulated fields with the ground truth for the validation simulation of the algorithm.

	Error AE	Error POD	Error kPCA
$\boldsymbol{q}$	$0.149 \cdot 10^{-4}$	$0.257 \cdot 10^{-4}$	$0.141 \cdot 10^{-4}$
$\boldsymbol{v}$	$4.1 \cdot 10^{-4}$	$19 \cdot 10^{-4}$	$6.25 \cdot 10^{-4}$
e	$0.472 \cdot 10^{-4}$	$0.64 \cdot 10^{-4}$	$0.342 \cdot 10^{-4}$
$\boldsymbol{\sigma}$	$5.1 \cdot 10^{-4}$	$20 \cdot 10^{-4}$	$3.371 \cdot 10^{-4}$
$\boldsymbol{\tau}$	$0.798 \cdot 10^{-4}$	$19 \cdot 10^{-4}$	$3.36 \cdot 10^{-4}$

Table 4.2: Loss comparison among SAE, kPCA and POD.

stress, and energy fields that cannot be measured with the camera, adding information to the user of a manipulation system.

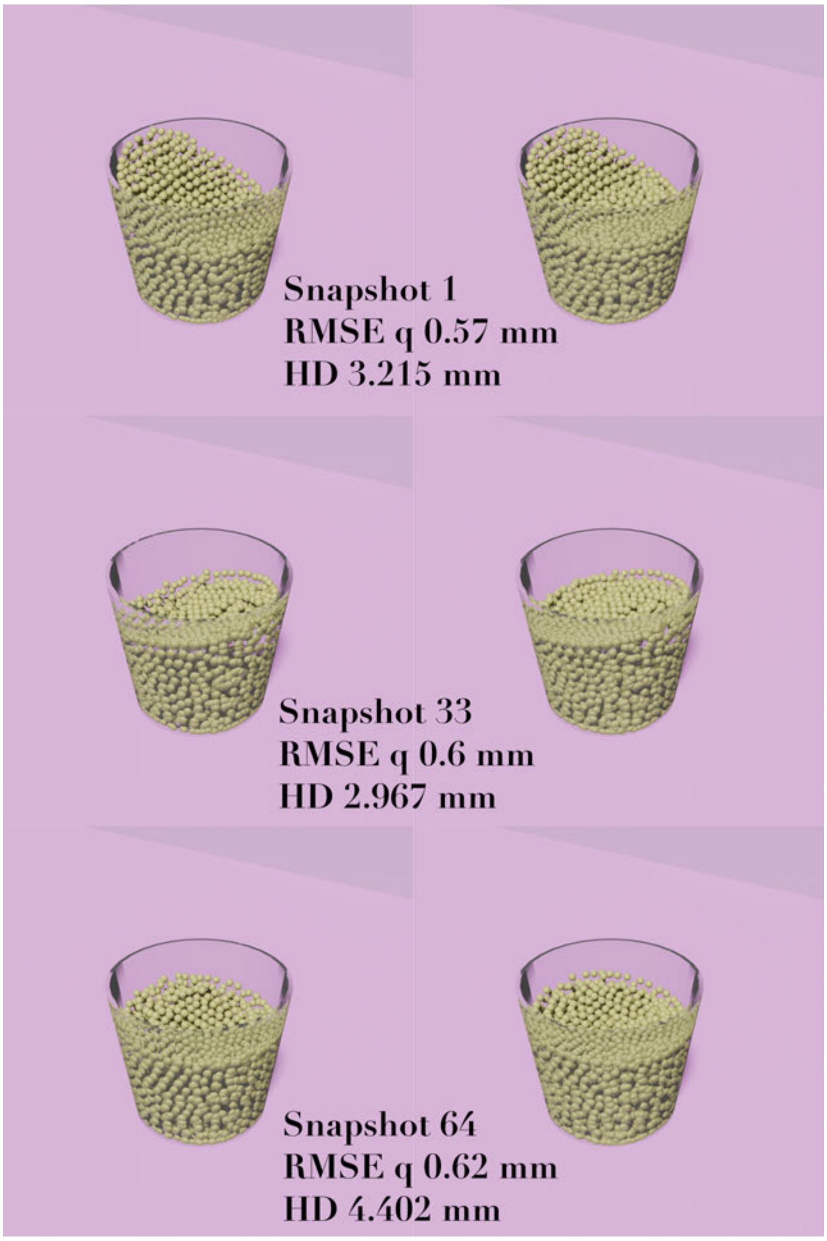


Figure 4.8: Comparison of the reconstruction of the integration provided by the SPNN (right) with the ground truth (left). The selected snapshots correspond to peaks of the sloshing dynamics of glycerine. Specifically, we present the comparison for snapshots 1, 33, and 64 of the collection. The height of the cup is 7 cm, and it is filled up to 5.6 cm approximately.

4.6.1 Computer vision

The first step is to develop a computer vision strategy that tracks the free surface and reconstructs the position of those points from pixel to real-world coordinates. In this case, we use the camera RealSense D415 (https://www.intelrealsense.com/

depth-camera-d415/), which also provides the extrinsic and intrinsic parameters. The camera outputs the depth field of the scene and the projection to 3D is straightforward. Nevertheless, the camera does not provide good enough measurements of the position of the cup and the liquid. Measurements related to transparent objects are often invalid or noisy since their surfaces are not Lambertian, which is the main assumption of the measurement algorithm incorporated in the stereo camera. In other words, instead of reflecting light evenly in all directions, they also refract light, resulting in unmeasurable conditions for the technique defined.

The camera provides depth estimation of the detected point cloud. The theoretical limitation in depth estimation of the camera at the operation distance is given by the following correlation for the resolution measurement (RMS) error:

$$\text{RMS}(mm) = \frac{\text{Distance}(mm^2)\ \times \text{subpixel}}{\text{Focal Lenght (pixels)} \times \text{baseline}(mm)},$$

$$\text{Focal Length(pixels)} \ = \frac{1}{2}\frac{\text{Resolution(pixels)}}{\tan\frac{\text{HFOV}}{2}},$$

where HFOV is the horizontal field of view. The majority of these parameters are given by the camera specifications, and the calculation of these parameters allows to evaluate the precision in the depth estimation. In well-textured targets, the theoretical error can be $RMS < 1$ mm. However, liquids and no texture scenes pose a problem for estimation that requires fine-tuning of the camera parameters.

Our approach consists in applying some filters to enhance depth streaming. Firstly, we apply a decimation filter to reduce the complexity of the measurements to foster stability. Then, the frame is mapped to a disparity map where the spatial filter, to preserve the edges, and the temporal filter, to promote data persistency, are applied. This result is projected back to the depth map where the hole-filling filter is finally applied. The filtered depth map outputs a full depth field from which we can evaluate the position of the features of the glass and the free surface (see Fig. 4.9). This procedure is fully detailed in the camera web documentation[1].

Once we have a continuous depth field, we detect and track the free surface. We convert the color image of the camera to black and white. Under appropriate fine-tuning of the conversion, the free surface can be detected as a gradient from black to white, as seen in Fig. 4.10. We alleviate the searching time of the gradient by defining an area for performing this analysis where the free surface is likely to be. The points of the free surface are detected, tracked, projected from frame coordinates to 3D, and stored for the metamodel.

4.6.2 Results

The video stream for validation consists of 800 frames, which is a recording of 12 seconds.

[1] https://dev.intelrealsense.com/docs/depth-post-processing

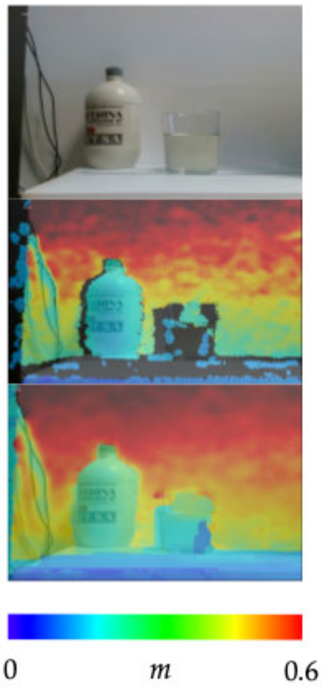


Figure 4.9: Color and depth stream before (up) and after (down) applying filters to reconstruct the depth map

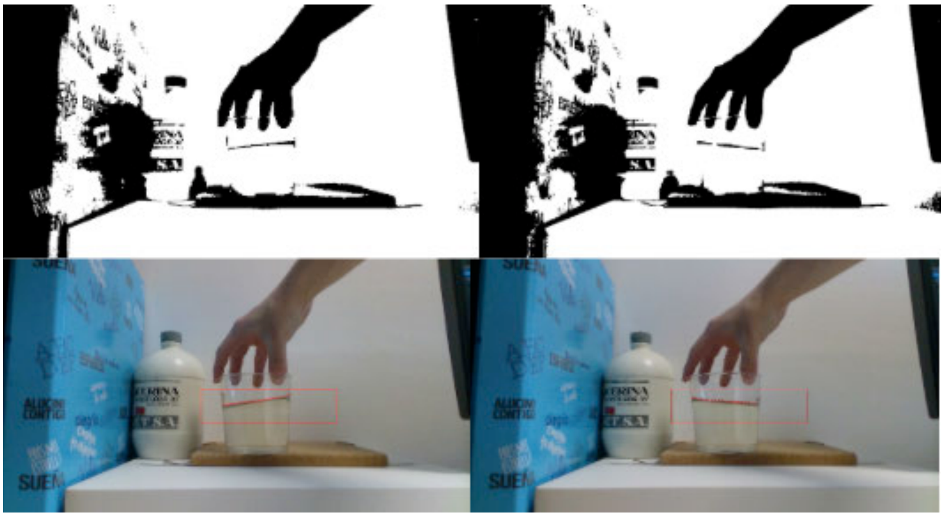


Figure 4.10: Representation of the color frame and its conversion to a binarized image to seek the free surface. The area defined for searching is represented in the color frame as well as the points of the free surface detected in the black and white image.

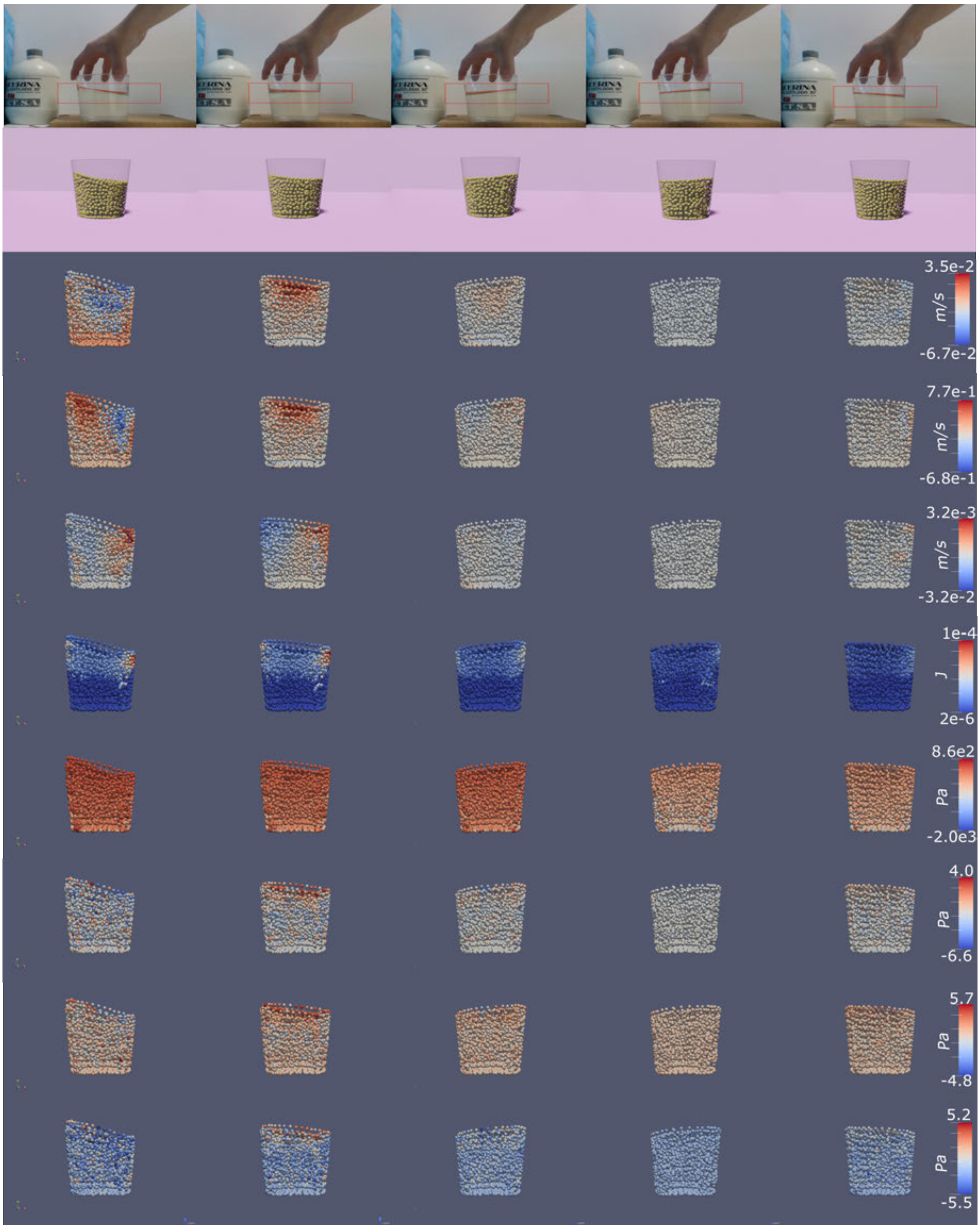


Figure 4.11: Results for a 12 seconds video of a glass of glycerine. Eight snapshots of the sloshing sequence were selected for comparison. The selected snapshots have indexes 560, 565, 568, 572, 578 from left to right. The second row corresponds to the fluid reconstruction and prediction provided in the previous snapshot. From rows three to ten we show the additional information obtained from the reconstruction and simulation (velocity, energy, and stress fields, respectively).

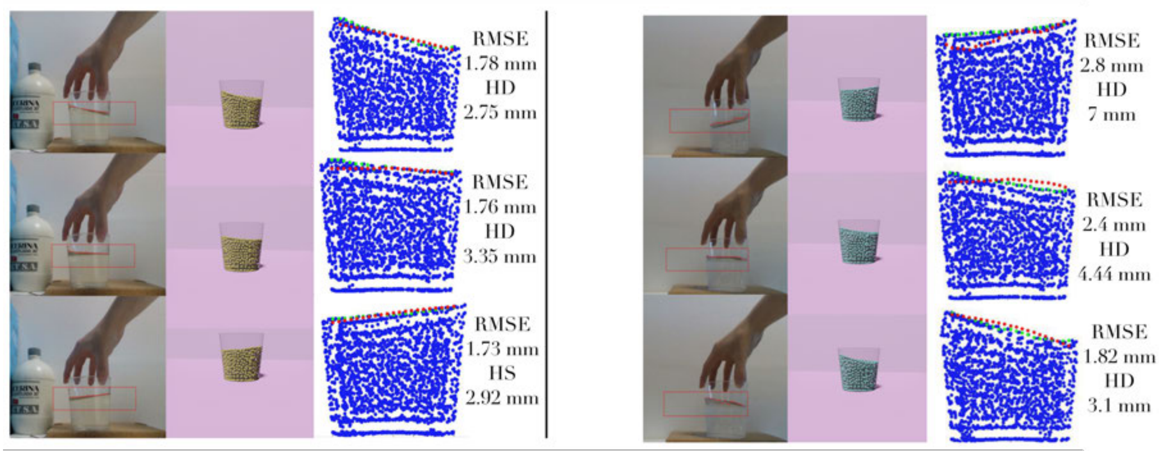


Figure 4.12: Detail of the comparison of glycerine (left) and water (right) with the prediction. The third column of both liquids compares the predicted fluid volume (in blue), the free surface of the liquid volume (green) and the target free surface (in red). The RMSE and the Hausdorff distance (HD) that correspond to each snapshot are indicated.

Fig. 4.11 shows the results of the algorithm compared to the real video streaming. We perform the reconstruction and integration over the whole sequence, i.e. no cuts were applied to the streaming and the method is applied continuously. We apply the three steps (RNN projection, integration, and decoding) over the full video in 3.42 seconds on an ordinary laptop (Macbook Pro 2013-3 GHz Intel Core i7), achieving (much more than) the real-time performance proposed. Some snapshots of the sloshing were selected and plotted in the first row of Fig. 4.12. The snapshots shown represent the peaks, which are the most critical states in manipulation, and some intermediate states between the peaks. The rest of the pictures correspond to the augmented information obtained with this method, which has been possible thanks to the physics-aware simulation framework.

All results of the integration are stable, realistic, and close to the real result. We analyze objectively the results by evaluating the root mean squared error (RMSE) between the real $\boldsymbol{y}$ and the predicted $\hat{\boldsymbol{y}}$ free surfaces in n snapshots of the video streaming. Ultimately, we feed the algorithm with the free surface in t (from the video), and we compare the integration result ($t+1$) with the free surface in $t+1$ (from the video),

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{t=1}^{n}(\hat{\boldsymbol{y}}_t - \boldsymbol{y}_t)^2}.$$

The evolution of the error along the video is represented in Fig. 4.13. The error remains under 5 mm in the whole sequence of the length of the video and stays lower than 3 mm in the vast majority of it. We also evaluate the HD between the free surface that comes from the camera and the simulation. These results reflect the closeness between the free surfaces, for which there is not a larger deviation than $4-5$ mm, even in the higher peaks of the sloshing. In some cases, higher deviations in the HD come from distortions in the detection of the free surface (like in the first snapshot of water presented).

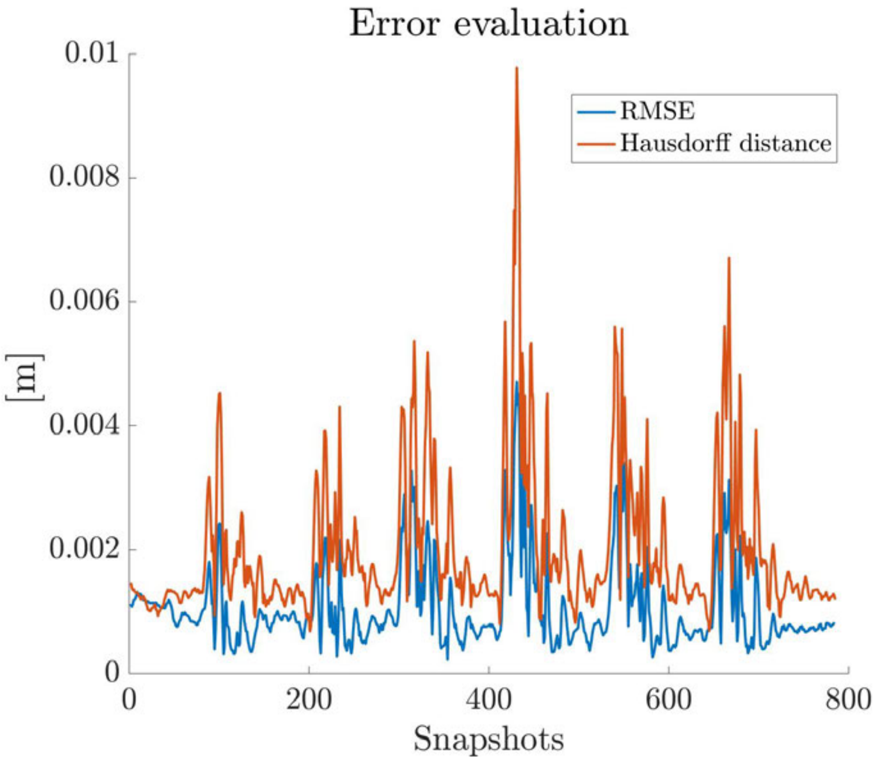


Figure 4.13: Evolution of the mean squared error during the perception process of sloshing in a glass of glycerine.

Fig. 4.14 showcases finally the compliance of the principles of thermodynamics in the predictions. The time derivative of energy makes little oscillations due to the numerical approximation around zero, which means that we ensure the conservation of energy. In addition, the time derivative of entropy remains always positive, fulfilling its production.

4.7 Conclusions

This theory opens another challenge: the smart data paradigm. We can see an opportunity in the abundance of data nowadays. We have information available from cutting-edge sensors and data acquisition systems located in any possible location. Conversely, the data required is not always accessible. In addition, gathering all the data required to build a sufficiently general and trustable system would be unaffordable. Physics knowledge and manifold learning are claimed as powerful solutions on this matter. Introducing physics priors as learning biases in machine learning has been a turning point for efficiently learning new models. Among the strategies to be followed, thermodynamics is presented as an appealing bias to adapt to different dynamical cases, including those that go beyond equilibrium.

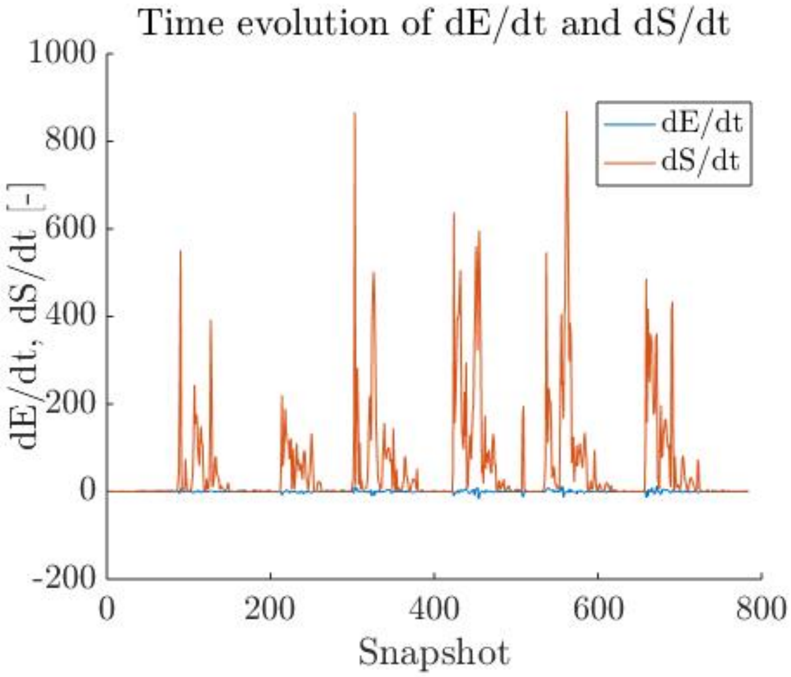


Figure 4.14: Time derivatives of energy and entropy along the video. The time derivative of energy oscillates around zero, ensuring energy conservation. Entropy production is also ensured since the time derivative is always positive.

Chapter 5

Perceiving and reasoning about previously unseen scenarios

In this chapter, we propose a physics-informed reinforcement learning strategy for fluid perception and reasoning from observations. Starting from full-field and high-resolution synthetic data for a particular fluid, we develop a method for the tracking (perception) and analysis (reasoning) of any previously unseen liquid whose free surface is observed with a commodity camera. This approach demonstrates the importance of physics and knowledge not only in data-driven (grey box) modeling but also in the correction for real physics adaptation in small data regimes with partial observations of the dynamics. The method here presented is extensible to other domains for the development of cognitive digital twins, able to learn from observation of phenomena for which they have not been trained explicitly.

The work presented in this section can be consulted in:

- Moya, B., Badías, A., González, D., Chinesta, F., & Cueto, E. (2022). Physics-informed Reinforcement Learning for Perception and Reasoning about Fluids. arXiv preprint arXiv:2203.05775.

5.1 Introduction

We have developed a system for physical scene understanding based on simulation with neural networks to establish a connection with the real fluid with measurements of the free surface. Now, we consider the case where the twin encounters a new fluid that has not been considered in the training database.

Despite the great advances achieved in the field of artificial intelligence, and particularly in the branch of physics perception, solutions are still designed for specific problems and restricted environments. However, there is still scope for the development of high-level reasoning skills for a machine to be able to adapt to new situations. For instance, considering this hypothesis, a machine would be capable of perceiving and reasoning about a new complex system with some knowledge previously acquired.

A first approach would suggest assembling narrow artificial intelligence instances that focus on little tasks. By combining them, a machine could be able to perform a wider range of actions to solve general problems. Nevertheless, a more natural approach consists in, given some piece of information, adapting to new environments and situations from the maximization of a reward that leads to the desired objective.

Reinforcement learning (RL) is the branch of artificial intelligence that focuses on learning from the machine-environment interaction. The main idea is to establish a correlation between states, such as the dynamical state of an element in the environment of the state of the agent itself, and actions that lead to the achievement of a certain goal. This mapping between the states and the action is learned as an optimization problem where we seek the maximum reward for a series of decisions. For this reason, reinforcement learning is considered a goal-oriented algorithm.

Model-based reinforcement learning relies on models of the physical world when the collection of data is difficult, or the required information is inaccessible [Wang et al., 2019a] [Ke et al., 2019]. This information is thus completed by a model that recreates the perception of real physics to make decisions about future events. Given the high complexity of the physics involved in real events, RL optimization usually relies on data-inspired modeling techniques of the dynamics to achieve high adaptivity and perform the optimization of the model and control policies all at once. Works include examples based on linear regression [Parr et al., 2008], the use of diverse neural networks [Hester and Stone, 2012] [Oh et al., 2015] and Gaussian processes [Deisenroth et al., 2013]. Conversely, current proposals introduce physics-informed machine learning into the model's inference [Liu and Wang, 2021].

Although physics-informed simulators reach higher generalization than unconstrained models, there are still difficulties to match the model outputs with an evolving real physical environment [Atkeson and Schaal, 1997]. For this reason, the learned simulator can be formulated as a reinforcement learning optimization problem to adapt to new scenarios through continuous observation. In addition, the imposition of inductive biases could be found advantageous in the correction process to reduce error bounds and adapt to new scenarios more efficiently [González et al., 2019a]. Examples of model discovery with reinforcement learning include the Burger's equation as a common benchmark [Benosman et al., 2021] [Bassenne and Lozano-Durán, 2019] [Wang et al., 2019b], or the Kuramoto-Sivashinsky equation [Bucci et al., 2019]. In this context, this strategy also targets turbulence and flow control problems, where the object of the optimization is also the physics simulator [Rabault and Kuhnle, 2019] [Ren et al., 2021] [Verma et al., 2018] [Garnier et al., 2021] [Novati et al., 2021].

In this chapter, we develop a method for tracking of the free surface and perception of the fluid and reasoning (providing the user with full-field information—velocity, stress,...) about the physical state of a sloshing fluid. While previous works employ simulation as the engine of physical scene understanding (see [Allen et al., 2020] or [Battaglia et al., 2013], for instance), thus needing previous knowledge of the physics of the scene and a pre-defined simulator, our method constructs the learned simulator on the fly. This learned simulator is built by resorting initially to synthetic full-field data, possibly coming from different fluids as described in Fig. 5.1. It then makes use of reinforcement learning in a thermodynamics-informed setting to correct systematic deviations of the observed reality from its predictions. Our approach ensures the compliance to first principles—conservation of energy, non-negative entropy production—of the resulting simulations, even if they are constructed from partial observations of the reality (in our case, the observation of the free surface of the fluid).

This intention is strongly aligned with that of hybrid twins [Chinesta et al., 2018]. If digital twins gap the frontier of simulation and reality to connect the machine and the real-world, hybrid twins lead to the self-correction of the inner model to optimize the accuracy of the solution in new scenarios that the user does not control. Both reinforcement learning and hybrid twins arise in response to the need for a smart data paradigm: profit data to overcome adaptation based on acquired knowledge to promote healthy use of data.

The development of this method revolves around the convergence of the aforementioned reinforcement learning strategy and transfer learning [Taylor and Stone, 2009]. Along the same lines of reinforcement learning, transfer learning profits from a model already learned to apply the gained knowledge to new tasks. This technique has also been considered in the a-posteriori correction of models such as manifold learning in reduce order modeling of fluids [Mohebujjaman et al., 2019] and dynamical optimization [Laroche and Barlier, 2017] [Goswami et al., 2020] [Guastoni et al., 2021].

The success of the proposed method is conditioned by the characteristics of the dynamics learned in the source model. If the features are sufficiently general, and the algorithm has correctly captured the patterns of the fluid dynamics, the reinforcement learning scheme will smoothly transition to the new perceived liquid. Therefore, this method relies on the capacity of GENERIC as an inductive bias algorithm to learn the major insights in data, regardless of the amount or the level of description of the data provided.

5.2 Method

In this section, the basic concepts of reinforcement learning and its application in model discovery are introduced for a correct understanding of the method proposed. A more detailed description of the method and different typologies can be found in [Sutton and Barto, 2018]. RL is described as a Markov Decision Process problem, and optimization

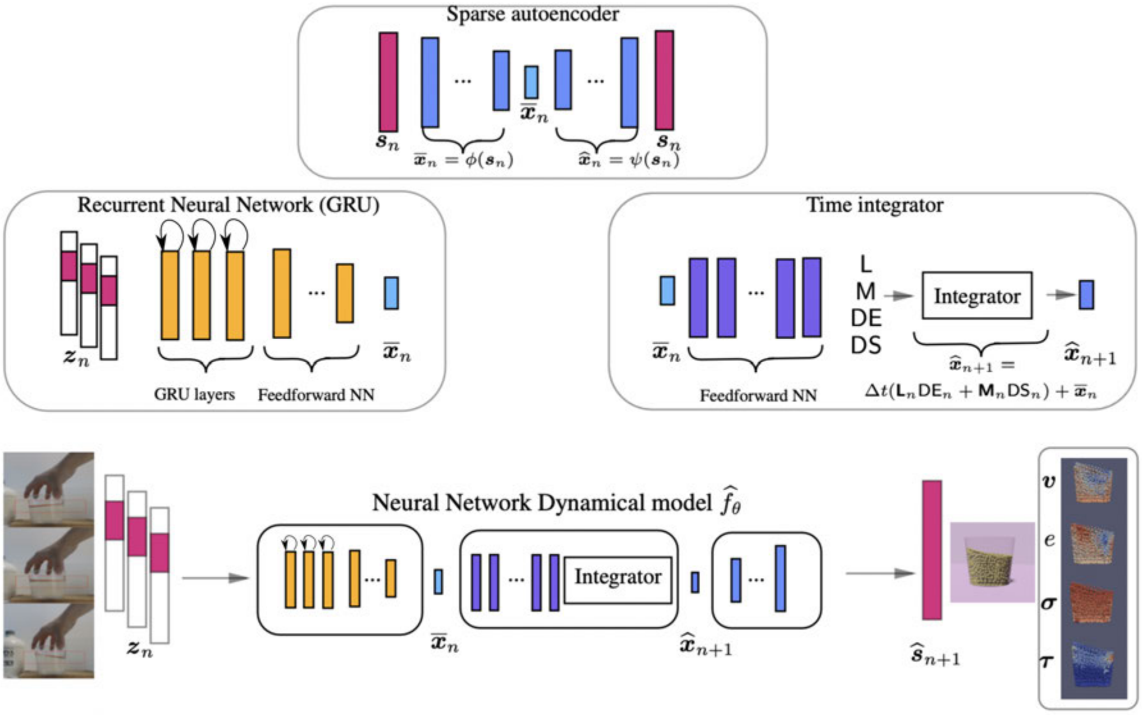


Figure 5.1: Original simulation engine trained for one liquid. It will be the starting point for the reinforcement learning algorithm.

focuses on learning explicitly a correlation between the actions and the states. This is known as a policy-based method.

5.2.1 Reinforcement learning background

Reinforcement learning is composed of certain elements that define the interaction and learning scheme of this approach. An agent, in this case, a computer, plans how to reach a goal in a potentially complex environment by learning to make sequences of decisions. The environment can be understood as a real scenario or a computationally recreated environment.

The reinforcement learning formulation is inherited from Markov Decision Processes, which proposes a modeling framework for some decisions to be made, where the evolution of the control object is conditioned by the actions of the user. Given the state $\boldsymbol{s}$ of a process at a specific time, the user or agent may choose an action $\boldsymbol{a}$ concerning the state of the process. The next step in the process state $\boldsymbol{s}_{n+1}$ depends on the previous state and the action chosen, but not on previous states and actions to $\boldsymbol{s}$ and $\boldsymbol{a}$. This condition satisfies the Markov property. The optimization is performed based on a series of states and decisions:

$$\boldsymbol{\tau} = (\boldsymbol{s}_1, \boldsymbol{a}_1, \boldsymbol{s}_2, \boldsymbol{a}_2, ..., \boldsymbol{s}_n, \boldsymbol{a}_n).$$

S stands for the state space and A is the action space of the process. r is the reward obtained from the transition from $\boldsymbol{s}_n$ to $\boldsymbol{s}_{n+1}$ given the action $\boldsymbol{a}$. The reward will prize the actions that result in better decisions to reach an objective. In classical reinforcement learning, we distinguish two main parts: the agent's policy and the reward function. The policy $\pi(\boldsymbol{s}, \boldsymbol{a})$ is the cause-effect relationship required to learn the control system, while the goal of the reward function is to, as a whole, retrieve the actions that result in more promising solutions driving towards the state to the system in the following time step $\boldsymbol{s}_{n+1}$. The goal is thus to find a correlation, or policy π, that reflects the interaction between the states and the actions. This loop is represented in Fig. 5.2.

In reinforcement learning, the transition probability P and total reward R are not known, and the agent must infer from the interaction the appropriate series of decisions based on an optimization theory. For the proposed case, the method employed is policy-based. In this framework, we develop an explicit correlation or mapping from the current state for the action to be taken. For this purpose, we formulate an objective function from the expected rewards of the decisions taken:

$$J(\theta) = \mathop{\mathbb{E}}_{\boldsymbol{\tau}\sim\pi_\theta}[R(\boldsymbol{\tau})],$$

where θ are the policy parameters that are optimized in the reinforcement learning algorithm:

$$\theta = \arg\max_\theta \mathbb{E}[R(\boldsymbol{\tau})].$$

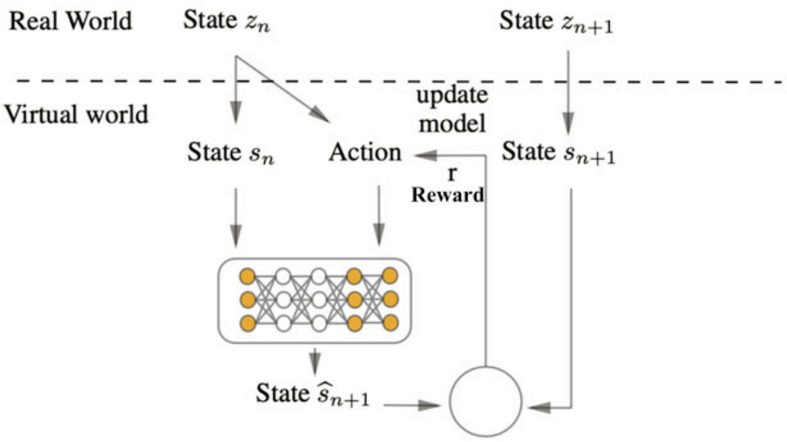


Figure 5.2: Reinforcement learning scheme. Given the partial observation of the free surface z_n at time $t = n \times \Delta t$ our method estimates the full state of the fluid $\hat{s}_{n+1}$. Its similarity with the next state s_{n+1} is evaluated to update the model. To preserve the patterns previously learnt, only a few layers are activated (in yellow) for backpropagation of the error and correction of the model.

5.2.2 Model adaptation from observation

The realization of this work requires the formulation of the proposed problem in the reinforcement learning framework. The methodology is applied to model discovery. Thus, the agent is the computer, or more precisely, the algorithm that we train, to learn a correct perception and reasoning model of the sloshing dynamics. The interaction with the medium is achieved by coupling the algorithm with a computer vision system that feeds it with observations. The goal of reinforcement learning is the optimization of the policy that correlates the state of the fluid with the actions. In this specific case, the states are the representation of the dynamics, and the actions of the model, or more precisely, the GENERIC parameters, that enable an accurate time integration of the slosh.

The sloshing dynamics is the real event of the environment we are interested in. It is described by its dynamical state $s_n \in \mathcal{S}$, where $\mathcal{S}$ represents the state space where the dynamics are embedded. Nevertheless, we only have access to a partial observation of that state $z_n \in \mathcal{S}$, which is the profile of the free surface. Those measurements are a partial subset of s_n. The free surface is detected and tracked with computer vision techniques that recreate the interaction with the environment.

The slosh then goes to the next state s_{t+1}. The actions a will lead to the prediction of the new state of the fluid. Specifically, the action refers to the proposal of the correct $\mathsf{L}_n, \mathsf{M}_n, \mathsf{DE}_n, \mathsf{DS}_n$ considering the current state of the dynamics. The policy $\pi(a|s)$ is the most convenient learned correlation to output accurate results that globally reward the reconstruction of the free surface from a recording of the slosh. The optimization is performed based on the parameters θ of the neural network.

The free surface is the object of study of liquid motion and, at the same time, the information available from the video recording of the real fluid. For this reason, the reward is formulated to guarantee its accurate reconstruction. The accuracy of the results is evaluated employing an L_2-norm. Since the dynamical insights of data, as well as the physical consistency, must be preserved during the correction, the reward includes a second term that penalizes the unfulfillment of the degeneracy conditions. The reconstruction is prioritized in the reward by including a weight factor in it, converting the degeneracy conditions into a soft constraint:

$$r_\pi = \lambda \frac{1}{N} \sum_N \|\boldsymbol{z}_n - \hat{\boldsymbol{z}}_n\|^2 + \frac{1}{N} \sum_N \|\mathsf{L}_n \mathsf{DS}_n\|^2 + \|\mathsf{M}_n \mathsf{DE}_n\|^2 .$$

We perform reinforcement learning as detailed in Algorithm 1. Each liquid has a dataset of N sequences available for training. We have a database of snapshots that we have assembled into N sequences for the RNN, which are associated to N snapshots of the state of the liquid. This information is used for updating the behavior policy. The process is repeated as many times as necessary to improve the accuracy and achieve convergence.

Algorithm 1 General Reinforcement learning pseudocode

```
Require: Free surface information $\boldsymbol{z}_n \in Z$ as a subset of the full dynamical state $\boldsymbol{s}_n \in S$,
  and the source model as initial behavior policy $\pi_\theta$
Ensure: Next dynamical states in time $\boldsymbol{s}_{n+1}$
  for Iterations until convergence do
      for n=1 to N sequences do
          Encoder $\overline{\boldsymbol{x}}_n \leftarrow \phi(\boldsymbol{z}_n)$;
          Compute forward propagation and determine action $a_n \leftarrow \pi_\theta(\overline{\boldsymbol{x}}_n)$, being $a_n$ the
  tuple $a_n = [\mathsf{L}_n, \mathsf{M}_n, \mathsf{DE}_n, \mathsf{DS}_n]$;
          Determine next integration step $\widehat{\boldsymbol{x}}_{n+1} \leftarrow \Delta t(\mathsf{L}_n \mathsf{DE}_n + \mathsf{M}_n \mathsf{DS}_n) + \overline{\boldsymbol{x}}_n$;
          Decoder $\widehat{\boldsymbol{s}}_{n+1} \leftarrow \psi(\widehat{\boldsymbol{x}}_{n+1})$;
      end for
      Compute reward $r_\pi$;
      Backward propagation and update RL policy $\pi_\theta$;
  end for
  return RL optimized policy $\pi_\theta$
```

In the reinforcement learning loop, we collect and provide data coming from observations to output the actions a with which we perform the time integration of the dynamics. The results are used to compute the reward that optimizes the network. Back-propagation is only applied to a selection of layers of the network of the source model. We not only alleviate the computational cost of the training but also ensure the preservation of the dynamical insights already learned in the source model to overcome the limitations of partial observations and low data regimes.

5.3 Results

5.3.1 Computational validation

The final goal is to develop the proposed methodology for real scenarios. However, we first test the method in a virtual environment with data coming from computational simulations. With this type of data, we will profit from precise error measurements to perform an initial evaluation of the scope of the method. The algorithm must correct the off-line trained source model for glycerine to adapt to new liquids with different properties and sloshing behaviors. The chosen liquids show different behaviors and rheology from higher to lower viscosity and density. Specifically, the three liquids selected are water, blood, and butter.

Training

The employ of pseudo-experimental data coming from simulations allows us to compute precise error measurements on the different adaptation strategies and their effect on diverse quantities of interest. Given a learned simulator trained off-line for glycerine data, we first apply the RL methodology to learn three different liquids presenting rather diverse behaviors: water and butter as Newtonian fluids, and blood, simulated as a non-Newtonian liquid. We employ synthetic data consisting in snapshots taken from four simulations for each liquid performed under different velocity conditions to trigger diverse sloshing results. From this information, we prepare the dataset with sequences of the positions of the particles that belong to the free surface. I.e., we track the free-surface particles, take their position, and prepare sequences of 16 snapshots for each time step Δt (the time instant of interest and the 15 previous states of the free surface). The length of 16 was found to be the minimum length of the sequences to find an embedding of the free surface measurements on each fluid's latent manifold, already computed off-line. The dataset of water has 750 snapshots in total, and butter and blood, 480. Synthetic data is available at a sampling frequency of 200 Hz, which stands for a time step of $\Delta t = 0.005$ seconds. Therefore, the source model was built for this availability of data and time step. Nevertheless, the snapshots are sampled at every frequency of 60 Hz, or equivalently $\Delta t = 0.015$ seconds, which matches the performance frequency of the commodity camera that will be employed in the real scenario. Hence, the change in the time step has also an effect on the model that should be corrected. The sequences of the dataset are randomly split into two subsets: 80% for training and 20% for testing.

We carry out the RL correction in the integration scheme and the embedding onto a lower-dimensional manifold. The correction is accomplished by activating the backpropagation in the last layer of the GRU network, and the 4 last layers, out of 13, of the SPNN. Since those are precedent layers of the decoder, and we train the network as a whole, we achieve the desired reconstruction with no need of altering the final structure of the network. The adaptation converges after 2000 epochs, at a small learning rate $lr = 0.0005$ and weight decay $wd = 0.00001$. We choose Adam optimizer [Kingma and Ba, 2014]. The reconstruction is weighted by a factor $\lambda = 2000$. Fig.5.3 shows the transition from the original manifold, trained off-line, to the new latent space that fits the emulated dynamics. Since the liquids bear a resemblance with the original liquid, glycerine, their manifolds do not show drastic changes in their structure compared to the initial solution. This fact also highlights the generality of the patterns of the dynamics learned in the source glycerine simulator.

Results

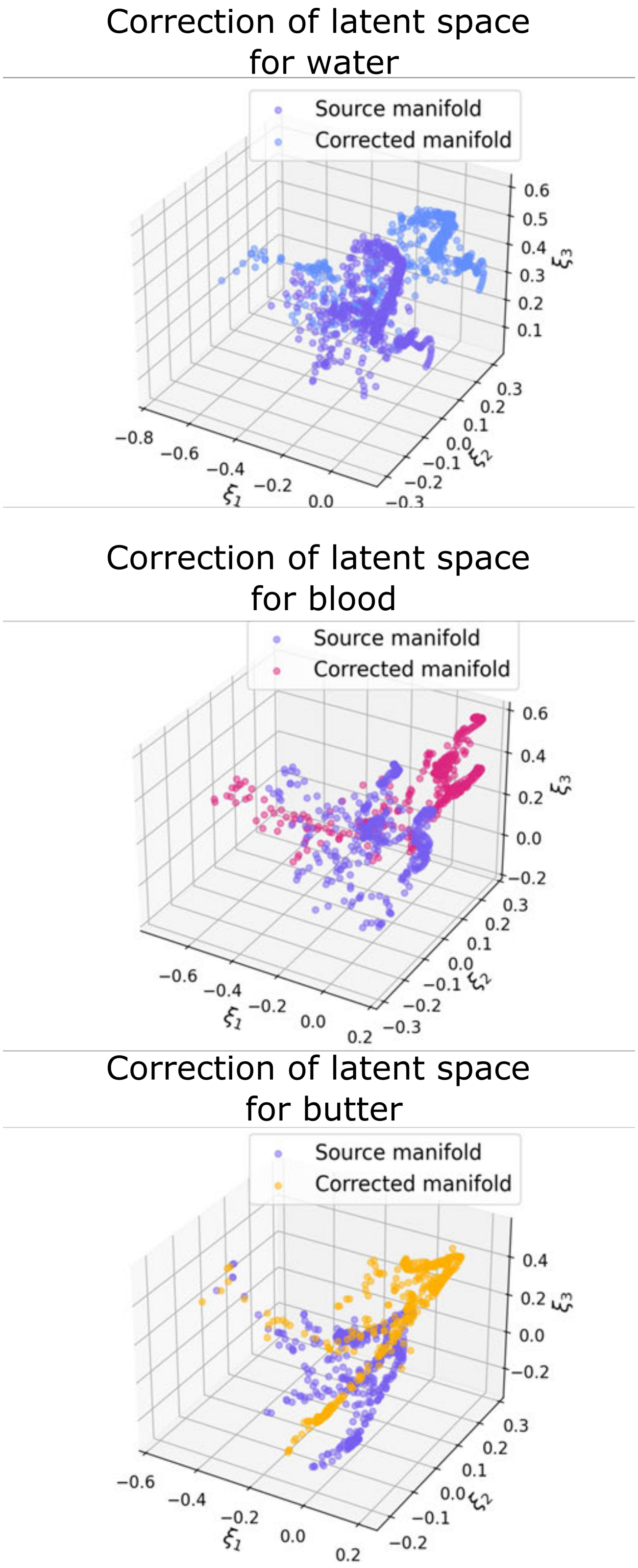
The proposed algorithm has achieved the desired correction of the liquids in the virtual environment. Fig 5.3 displays the changes in the low dimensional manifold where the dynamics are embedded. Since the liquids bear resemblance with glycerine, the liquid employed in the source model, the new manifolds do not show abrupt changes compared to the manifold that would be obtained with the source encoder. This result also emphasizes the generalization achieved in the source model through the patterns learned.

The main objectives to achieve in the adaptation are the improvement of the reconstruction of the free surface, included in the reward function, and the reconstruction of the maximum height, in particular for control systems that seek sloshing-free solutions. The coordinate system to fix the height reference is placed at the bottom of the vessel. Hence, the relative error is computed around this reference. We detect the points of the free surface to interpolate the profile of the free surface divided into equally spaced control points to calculate the error:

$$\text{error} = \frac{1}{N}\sqrt{\sum_{n=1}^{N}\frac{\boldsymbol{z}_n^2 - \hat{\boldsymbol{z}}_n^2}{\boldsymbol{z}_n^2}},$$

N being the number of samples of the dataset, and $\boldsymbol{z}_n$ and $\hat{\boldsymbol{z}}_n$ the ground truth and simulated free surfaces respectively. This error is displayed in Fig. 5.4. We appreciate an improvement in the reconstruction error of the free surface that showcases the performance of the corrected network. We present the reconstruction error for each liquid considering the full dataset. In other work, we evaluate the reconstruction for the snapshots of the training and test subsets together. We can distinguish the four simulations of each database in the graphs. The higher error occurs in the peaks of the slosh, where the deformation of the profile is more pronounced. However, the error remains in low ranges.

Water is less viscous than glycerine, and thus its slosh is higher and out of the ranges of the source model database. For this case, the maximum error drops from up to 12% to $4 - 6$%, remaining at an average of no more than 3%. The reconstruction error for butter is reduced from 6% to less than 2%. Finally, the algorithm adapts to the non-Newtonian description considered for blood, where the maximum relative error goes from 7% to 3.5%.



Correction of latent space for water

Correction of latent space for blood

Correction of latent space for butter

Figure 5.3: Representation of the correction of the latent manifold of the position. The latent representation evolves to match the features of the new liquid.

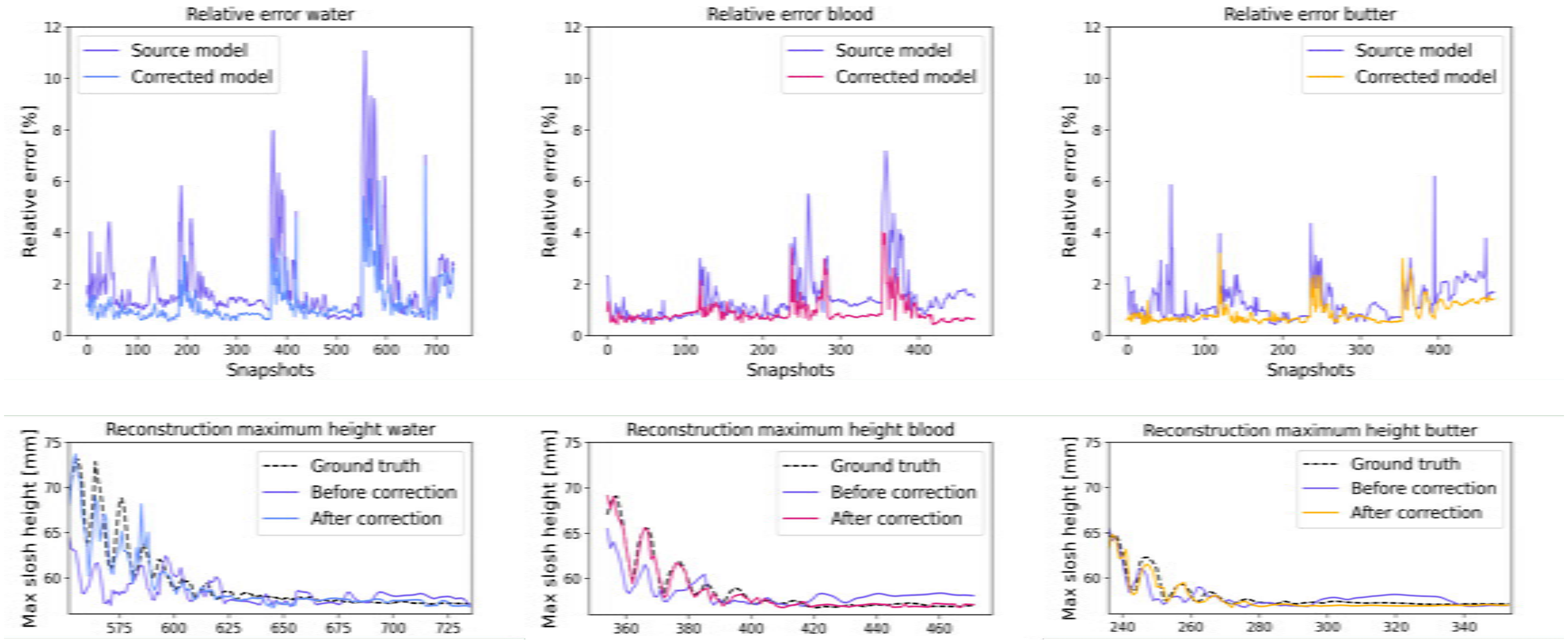


Figure 5.4: (Up) Relative error of the reconstruction of the free surface for the four simulations used for training and test. The relative error obtained with the source model is reduced after the optimization. (Down) Detail of the sloshing height reconstruction of the most critical simulation. The method correctly emulates the behavior in magnitude terms and, notably, in the precise time occurrence of the peaks.

Some small errors come from the interpolation of the free surface profile. We employ a meshless method and, despite the resemblance of the reconstruction, the differences between the ground truth and simulation particle distributions result in slightly different interpolated free surfaces. This could be improved by smoothing out the profile, considering a different interpolation method, or a finer selection of points. However, given the current real resemblance and low errors, we do not consider the application of these techniques.

The maximum height of the free surface is one of the objects of study of sloshing free control problems. For this reason, we also analyze the performance of the correction by evaluating the similarity of the sloshing peaks with the maximum height of the points of the free surface between the simulation and the ground truth. The graph has been augmented to properly appreciate the reconstruction of the highest point during the dynamics while preserving as much as possible the length scale without distortion. The amplitude of the slosh before correction seems dimmer than the ground truth. Conversely, the correction replicates the real, and higher, amplitude, as well as the time occurrence. Before the correction was applied, some peaks appeared delayed or did not appear at all.

Fig 5.5 shows a qualitative representation of the results obtained. This demonstrates that, after applying reinforcement learning, we maintain the shape of the fluid volume and it can be recognized as a liquid. In the correction, we could risk the shape of the fluid by imposing hard conditions in the reward. However, the whole fluid volume adapts to the new free surface.

5.3.2 Testing in real scenarios

The ultimate goal of the present work is the adaptation and recreation of the dynamics of real liquids. In this real-life environment, the algorithm has to learn models to mimic previously unseen liquids of different characteristics, properties, and thus dynamics, whose behavior is reflected in different amplitude and frequencies of slosh. Information comes in form of images from the video recordings. These images were employed to detect and track the free surface of real fluids employing computer vision to perform the correction and posterior validation of the method. Data is acquired with depth evaluation of the points of the free surface, which are detected from the black-white gradient in binary images, see Fig.5.6.

This problem showcases various challenges such as errors in depth estimation, the difficulty to detect and track the free surface, the complexity of real liquids to capture all the dynamical features from the recorded videos, and user actuation. The last statement refers to the direction of the movement of the liquid. In this work, we limit the experiment to a plane movement in two possible directions (move the glass to the left or the right), but we can experience slight deviations in the actuation. We impose this restriction to bound the problem to a case where we can evaluate properly the free surface from a fixed position of the camera with the detection method employed.

Information about the free surface is acquired unevenly. As represented in Fig. 5.7, we perform interpolation of the given data to have information of the free surface at equally spaced points.

Training

We evaluate four real liquids to evaluate the scope of the method, presenting different properties and behaviors. We have considered water, honey, beer, and gazpacho (a typical Spanish cold soup made of tomato). These are daily liquids that can be found in an ordinary routine, and thus their importance. Although water properties are widely known, the other three liquids are subjected to the production process and small variations in their composition result in slightly different dynamics. In addition, the changing environmental conditions can affect their properties. The liquids are contained in the same glass presented in previous sections, and it is filled to the same level approximately. The geometric changes of the liquid are not considered in this application.

We perform two recordings per liquid. One is used for training, in which only 80% of its snapshots are used for this purpose, and to avoid overfitting. The second recording is used for validation of the learned correction. Reinforcement learning is applied individually. For each liquid, we initialize the algorithm by transferring the information of the source model and start the correction with new observations.

Similar to the previous computational case presented, we apply transfer learning by deactivating back-propagation in the model to only allow it in selected layers of the model. We activate the last layer of the GRU network and 5 layers at the end of the SPNN to perform the correction. Here we had to activate one more layer in the SPNN to achieve sufficient adaptation and reduction of the error. This problem, due to its increased complexity, required more flexibility. On the other hand, we still had to ensure the preservation of the features of the dynamics learned previously. We complement the design of the methodology with a small learning rate, $lr = 0.0001$, and weight decay $wd = 0.00001$. The reconstruction contribution in the reward is weighted by a factor λ equal to 2000. Like in previous pieces of training in this dissertation, the optimizer employed is Adam.

Results

Fig. 5.8 showcases the results of the correction in training and test recordings of the four liquids. The four exhibit an improvement in the sloshing reconstruction compared to the performance of the source model before correction. In addition, the temporal integration with data from the test datasets presents a noteworthy performance considering that this information is new to the network. The training recordings are shorter than 10 seconds, and only three or four sloshes are captured in these datasets. However, the new model learns the new target behavior. Even though we work in a low-data regime to perform the correction of complex liquids, the method successfully reproduces the training and

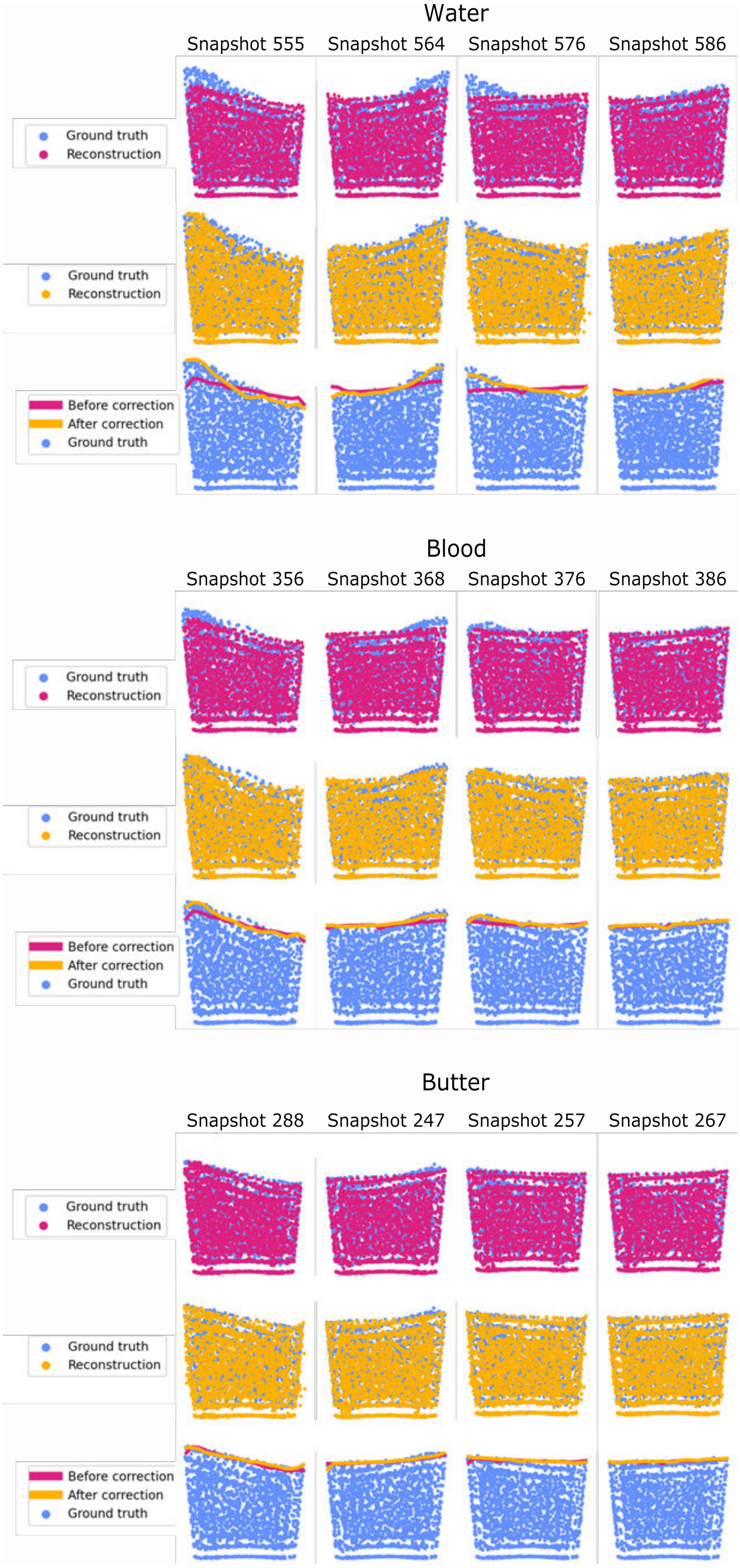


Figure 5.5: Reconstruction before and after correction of water, blood and butter.

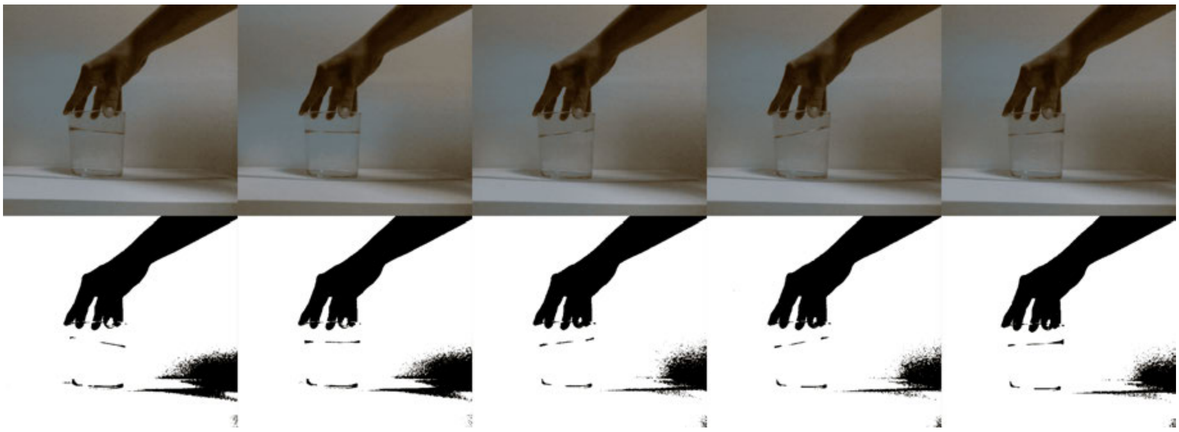


Figure 5.6: Representation of surface tracking. Each color frame is converted to a binary frame where the surface is detected (shown in red in the correspondent color frame). The depth map is built upon these points. For transparent liquids, like the glass of water shown in the picture, the depth map is of lower quality resulting in an incomplete detection of the surface.

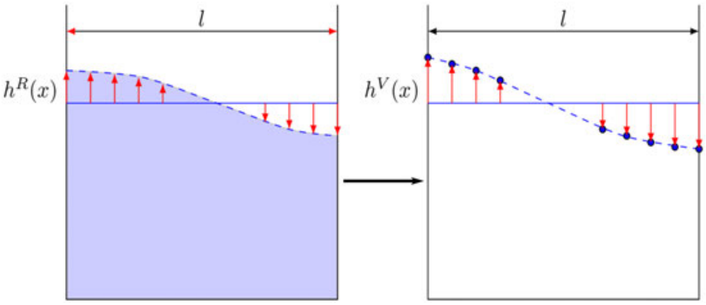


Figure 5.7: Representation of the interpolation method for data acquisition. Given some detected points from the free surface, we interpolate the height at equally spaced locations to describe the free surface in an homogeneous reference to facilitate comparison and learning.

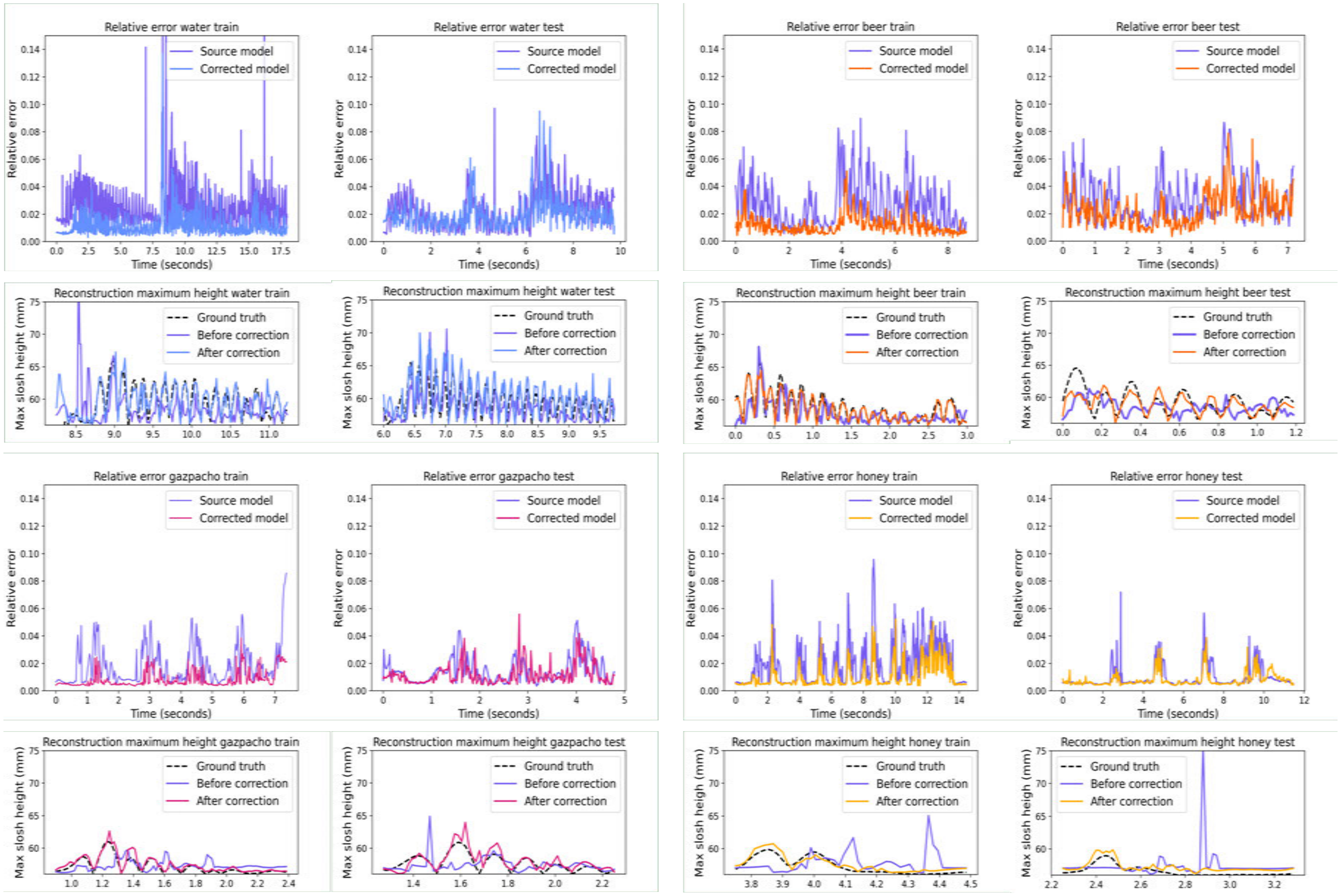


Figure 5.8: Relative error and reconstruction of the sloshing height for train and test recordings of water, beer, gazpacho and honey.

test benchmarks. These results are obtained because of the inductive biases learned and preserved in the algorithm, leading to an efficient correction from limited data and partial measurements, in this case only considering the free surface for the reconstruction. Water and gazpacho present slightly higher errors than the other liquids in the test dataset. This is probably due to the difficulties experienced in the data acquisition and the noise of the samples in the experiment. Additionally, water has higher slosh frequencies and longer slosh time. Therefore, more varied sloshes would be needed to reduce the error, still being a low-data regime. Despite this, the method already performs the adaptation over train and test information correctly. Fig. 5.9 represents the reconstruction of state variables during simulation. The algorithm employs information of the free surface to correct the simulator and perform the simulation of the dynamics in time. The simulation is performed in the low dimensional space with the latent representation of the whole set of state variables that GENERIC employs for the description of the fluid evolution. As a result, the algorithm additionally outputs a reconstruction of these state variables (velocity, stress, and energy fields). Finally, Fig. 5.10 shows renders of the simulation results before and after correction compared to the snapshot that they predict in time. The time is indicated to correlate each render to the error in Fig. 5.10. Despite the correction, the algorithm outputs a matching shape with the real entity. As a result, giving only the free surface of the observed fluids, the networks integrate in time the dynamical evolution of the liquid and provide a three-dimensional reconstruction of the fluid volume. The particle discretization represents a fluid volume that can be translated to the user using renderization and augmented reality. This representation bridges the gap between reality and the virtual environment to provide augmented information to the machine, and the user, for decision making.

5.4 Conclusions

The present chapter shows a reinforcement learning methodology guided by GENERIC as an inductive bias for perception and reasoning about fluid sloshing. The algorithm learns from observations to accurately mimic new fluid behaviors from the sole observation of the free surface. The method provides a tool for model inference with real data from partial observations of complex dynamics. The correction of physics perception enables the machine to adapt and learn previously unseen liquids present in daily tasks with unknown properties. We start from a source model trained with computational data to learn a physically sound simulator of the sloshing dynamics upon GENERIC to ensure the physical consistency and generalization of the results. The calculation is performed in a low-dimensional manifold to ensure real-time performance. Thus, we obtain real-time interaction with the environment in which the model, or digital twin of the real liquid, operates to have response capacity.

We illustrate the benefits of physics-informed deep learning for reinforcement learn-

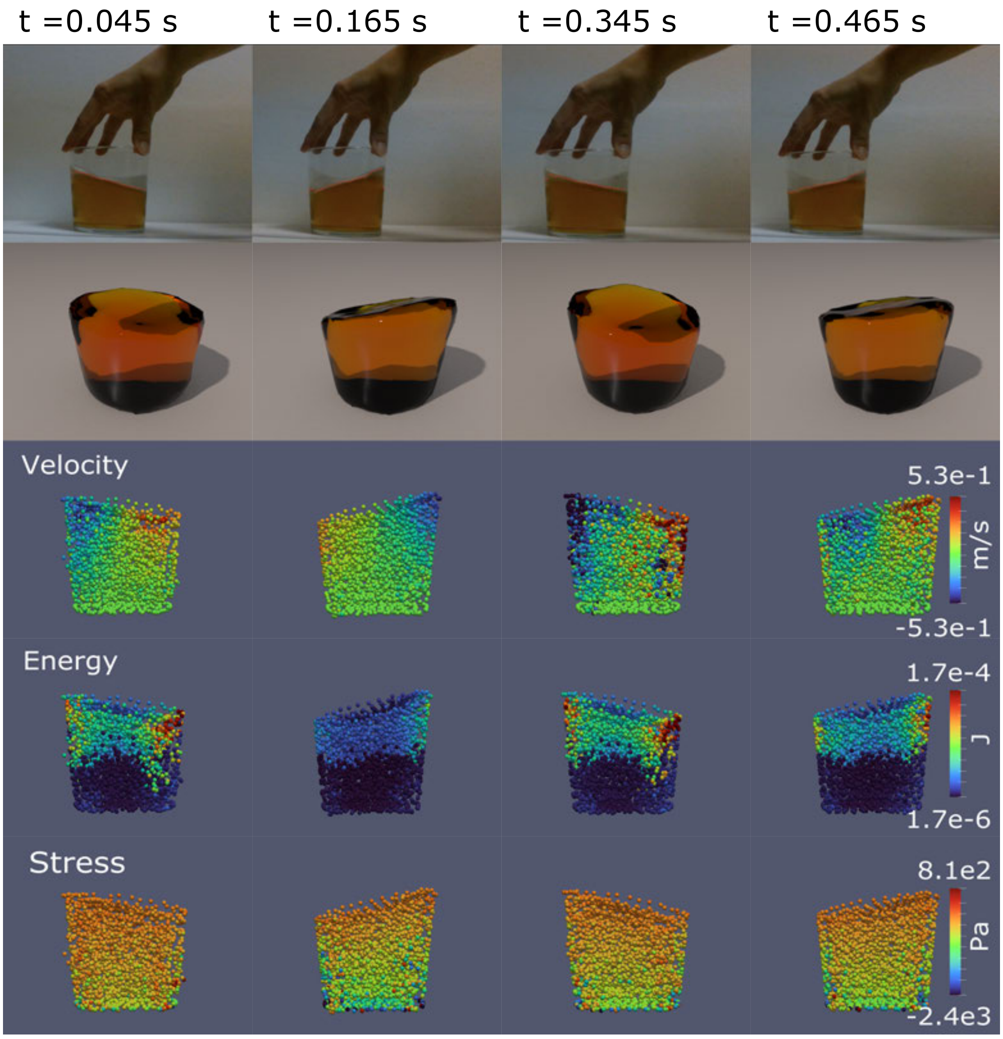


Figure 5.9: Reconstruction of state variables during time integration for beer. The reconstruction of the vertical velocity, energy and normal stress of the particles is shown at four integration snapshots

ing and correction. Given an off-line learned simulator for one particular fluid (glycerine, in our case), our method manages to evolve to a new representation of the dynamics to match the behavior of previously unseen liquids. The recordings employed are limited to 10 seconds approximately. In this low-data regime, the method adapts to the new dynamics perceived. The good performance of the method can be observed also in the test recording, which has not been seen by the network before. The success is attributed to the insights learned in the source model with simulation data, and the inductive bias imposed by GENERIC, which ensures the fulfillment of the principles of thermodynamics. These are sufficiently general and precise to allow the simulator to evolve smoothly to new liquids.

A challenge in physics perception is the balance between adaptivity and the risk of learning noise coming from the experimental nature of the data acquisition technique.

Liquids and vessels are non-Lambertian, i.e., they do not have a diffusely reflecting surface, or matter, convenient for depth estimation. Despite the fine-tuning of the camera, the measurements include noise and invalid measurements from which the free surface has to be reconstructed. By applying transfer learning and performing slow training we have prevented the network from learning meaningless information coming from measurements. Moreover, the patterns already learned help reconstruct the information to learn the new behaviors accurately.

Despite the accuracy observed in the reconstruction of train and test recordings, the performance of the method could be further improved over the last model learned by retraining with new datasets acquired by the same means. Hence, it can perform corrections to not only persist in the improvement of the reconstruction but also to adapt to the evolving nature of the scenario.

The results observed can be a starting point to adapt the method to new geometries and include this new parameter in the optimization from the application of geometric deep learning. In addition, this problem could be further extended with more general liquid detection techniques capable of analyzing transparent and not textured elements from different perspectives [Sajjan et al., 2020] [Do et al., 2016].

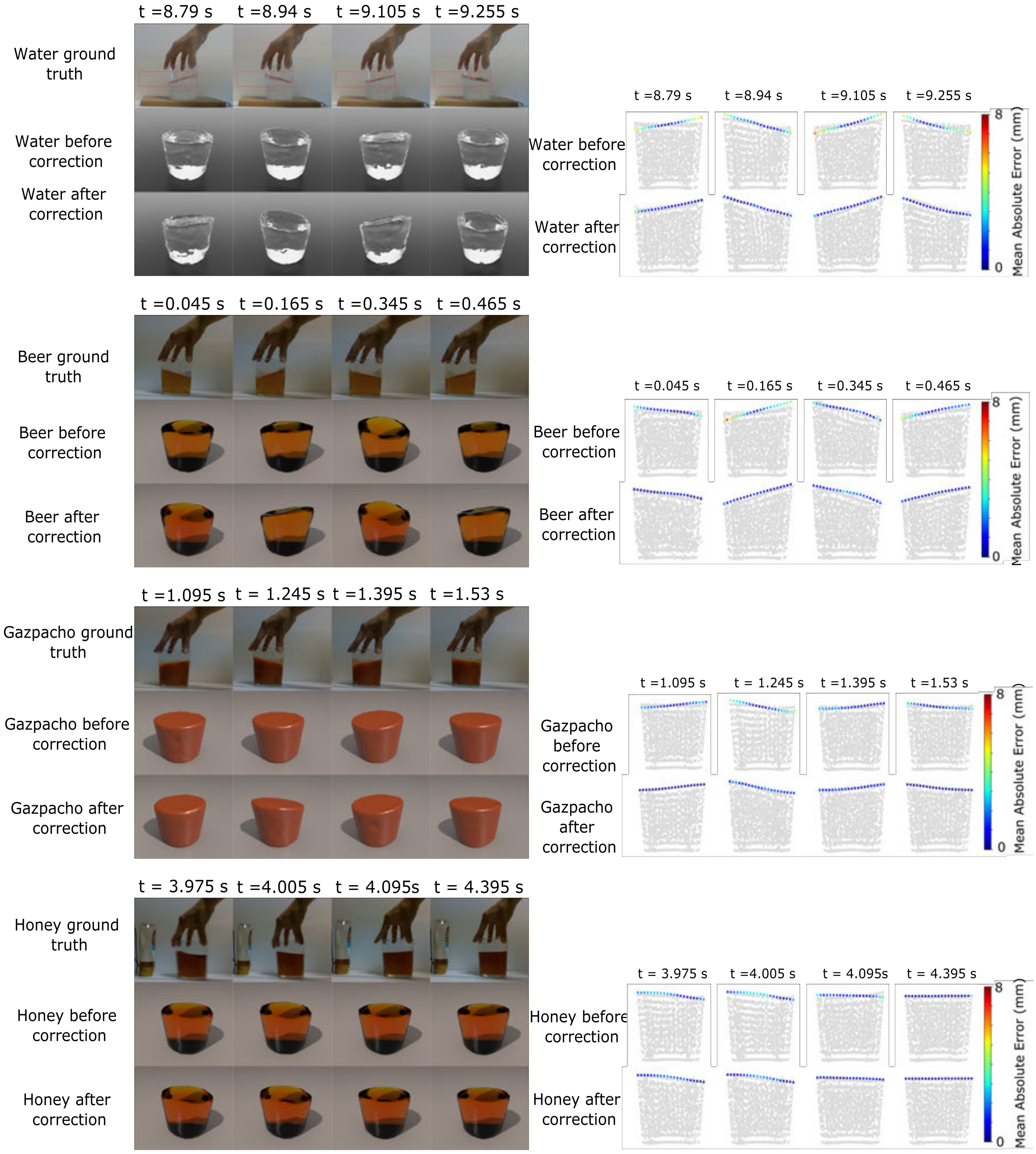


Figure 5.10: Render of the volume reconstruction before and after correction. The algorithm outputs a particle discretization of the fluid that can be presented as a three-dimensional render of the volume for visualization and interpretation. The figure shows the peaks of the dynamics observed in a piece of the recordings.

Chapter 6

Conclusions

6.1 Introduction

This chapter of the dissertation intends to provide an overview of the main results of this work that leads to their main conclusions. In addition, a review of future research lines will be presented, as well as the dissemination plan followed and complementary material and activities of the candidate to give a full view of the work done along the research process.

6.2 General conclusions

The dissertation provides a framework for physical scene understanding, focused on the sloshing dynamics of liquids. It states a challenge in both computational and computer vision fields due to its complexity. We set the basis of a data-driven approach guided by physics to describe the dynamics and learn meaningful insights for a deeper understanding of the phenomenon that will enable reasoning and decision-making with data coming from observations.

The thesis has explored the different steps of the perception and reasoning loop. Starting from learning a model of the dynamics in which we are interested, we have progressively evolved towards the development of a digital twin of a real liquid that provides an interpretation of the real dynamics with small and limited data. Finally, the digital twin became a hybrid twin, a system capable of correcting and adapting to new behaviors from its experience to learn with limited data. It is worth noting that this work was conceived and envisioned from the field of computational modeling and mechanical engineering, and it aimed to propose a solution combining methods from different fields to create a global solution for the perception problem proposed.

We deeply explored machine learning techniques to cover a far-reaching set of applications to work over the stages that form perception, which includes simulation and prediction, data acquisition and interpretation, and reasoning and knowledge application.

The project revolved around supervised learning to prove the effectiveness of the GENERIC formalism as a bias for model inference. First and foremost, this approach guarantees compliance with the conservation of energy and entropy production in the algorithm where it has been implemented. This characteristic makes it suitable to learn descriptions of complex dynamics with unresolved degrees of freedom and inherent dissipative nature. Notably, in the case of learning from measurements of the real world, we cannot rely on *perfect* approximations that assume no dissipation. Secondly, GENERIC can be obtained numerically, and the flexibility for its implementation in regression problems is convenient.

It is worth mentioning that it has been employed in two different approaches for learning the behavior of the sloshing dynamics. First, we follow the approach of *slow manifold* construction. Secondly, we learned a correlation with ANNs. Third, it shows a good response for its implementation in conjunction with model order reduction techniques. Finally, the numerical discretization by finite differences is not severely conditioned by the time step employed in the discretization as long as it samples correctly the manifold of the dynamics.

Manifold learning was satisfactorily applied for sloshing understanding. However, ANNs provided a higher degree of generalization of the method. Then, instead of performing interpolation in submanifolds of the manifold to perform the integration with the interpolated values of the GENERIC elements, this process is learned and optimized with neural networks. This approach was especially convenient for correction since we performed the optimization directly over the approximated function of the source model.

Model order reduction, considered part of the field of unsupervised learning, was a key asset in the development of physics simulators. The particle discretization was favorable for data extraction and computational simulation, but MOR extracted the features underlying the movement of the particles to achieve a deeper understanding of the dynamics of this phenomenon. Results supported the use of non-linear model order reduction techniques to learn the non-linearities that characterize the sloshing dynamics. POD required a higher dimensionality to define the low dimensional manifold than LLE, TDA, k-PCA, and autoencoders with all the features of the dynamics necessary to learn GENERIC and extrapolated to simulations out of the database.

Autoencoders were chosen in the last steps of the implementation for consistency of the ANN approximation proposed and also to perform correction in the reinforcement learning loop. It achieved good performance compared to TDA and k-PCA. TDA, although accurate, required more computational resources than the other two methods. In addition, the autoencoder worked over each set of state variables (position, velocity, energy, normal stress, and shear stress), capturing more information of each group individually. A final argument in favor of autoencoders is their speed of computation. The method proposed in the second chapter fulfilled real-time constraints performing a simulation of 1.7 seconds in 1.6404 seconds. However, the method developed with ANNs performed a simulation of 12 seconds in 3.42 seconds, outperforming the previous approach.

MOR techniques were also exploited for manifold learning of fluid dynamics. k-PCA unveiled a 3D low dimensional manifold where dynamics were embedded. We discovered the liquids to be clustered and following a structure as a function of their properties and the dynamical state. This result was the inception of the development of other applications, such as classification with a structure of random forest.

Data acquisition established the connection with the real physics that was to be emulated. The proposed algorithms were successfully connected to the real glass and the real liquid entity, overcoming measuring limitations with appropriate fine-tuning of the camera to perform not only the integration of the dynamics but also the correction of the emulation in a reinforcement learning loop.

Access to data can be expensive or not possible. In the case of this thesis, the camera provided only evaluations of the free surface but there was no option to measure the energy, for instance. However, a specific set of state variables was required to define the GENERIC of the sloshing dynamics. Self-supervised learning offered a framework to conceive the ideas to develop a method to reconstruct information. We hypothesized the possibility of learning the features of the dynamics from the evolution of the free surface to recover the dynamical state of the fluid.

Reinforcement learning has been applied in the last step of the cognitive twin to achieve the highest degree of adaptivity and reasoning. Just like robots learn actions based on reinforcement learning, reasoning can be trained from a reward to approximate our goal: a good approximation of the prediction of future dynamics.

Here, two main challenges were addressed. First, the source model to perform adaptation must have learned sufficiently general features to evolve towards the new solution. If not, reaching a new solution in the re-training will be inefficient, and arduous. In this case, we proved the benefits of model order reduction and learning biases in the context of reinforcement learning. The cognitive digital twin had been previously developed upon a known reality, which is the dynamics of glycerine. We proposed transferring this knowledge for learning new fluids, where the process is conditioned to the insights learned in both the low dimensional manifold and in the integrator. Finally, the results supported the coherence and robustness of these methods for a smooth transition.

Secondly, reinforcement learning by itself could not be enough for optimization. If it has to guide the evolution of the optimization, the reward has to be explanatory to be meaningful. In other words, in our case the reward must ensure that the algorithm learns to reproduce the evolution of the free surface and the physics, fulfilling the laws of thermodynamics. Although the reward is focused on the reconstruction of the dynamics of the free surface, we also proposed to define the reward with an additional term to maintain the thermodynamical compliance of the description. In this way, the structure of the reinforcement learning algorithm becomes more relevant in the optimization of an optimal solution. As a result, we successfully trained cognitive hybrid twins that learned from a short video sequence a new behavior that they were able to reproduce with new data, improving the performance compared to the source model. In addition, it reconstructs an approximation of the solution field (velocity, energy, and stress) for any fluid manipulated.

6.3 Thesis contributions

This work contributed to the development of a fully operative perception system capable of predicting dynamics in real-time and mimicking previously unseen fluids with knowledge of physics. In addition, the simulation is completed with the reconstruction of solution fields when only partial measurements are available for any known or new fluid. However, this major contribution has been possible with the accomplishment of specific tasks:

- Exploration of manifold learning techniques applied to the slosh for regression problems. POD has been compared with LLE, TDA, k-PCA, and autoencoders proving the efficacy of non-linear MOR for capturing the insights of the dynamics.
- Development of a model order reduction method based on TDA. This method has been implemented to propose an approach to preserve the topology of the full order manifold in the low dimensional representation.
- Application of machine learning methods based on the GENERIC formalism in regression for fluid dynamics simulation. This formalism has been employed in two approaches: manifold learning, and ANNs.
- Preparation of a tracking system for the container of the liquid and coupling to the simulation algorithm.
- Digital twin implementation and visualization of the solution in real-time by means of augmented reality to provide interpretation of the results and connection between virtual and real worlds.
- Study of a low-dimensional manifold of sloshing dynamics for a set of fluids. Exploration and analysis of clustering properties, and training of a classification system to distinguish fluids.
- Preparation of a data-acquisition system for free-surface detection, tracking, and depth estimation.
- Development of a RNN strategy for connecting partial measurements with the low dimensional manifold of the sloshing dynamics. Provided this projection, the full solution field of the fluid can be reconstructed in the current of next time steps.
- Development of a RL loop for correction and adaptation to new, previously unseen, fluid, with information of the free surface. A posteriori reconstruction of an approximation of the full solution field (velocity, energy, and stress) for new fluids, simulated in time.

6.4 Future research lines

The present work proposes advances in the field of fluid perception and reasoning. However, the exceptional advances experienced in the research community make patent the scope for enhancement of these systems. The proposals can be grouped into three categories concerning the three fields merged in the thesis.

6.4.1 GENERIC into learning

The implementation of GENERIC as a learning bias arouses curiosity in the community due to the characteristics highlighted before. However, its implementation can be seen from different perspectives, as seen in [Zhang et al., 2021] [Lee et al., 2021], where authors show alternatives for formulating the degeneracy conditions into the networks. In our approach we consider the application of the degeneracy conditions as a soft constraint convenient for training, assuming a low numerical error during this process.

Sobolev learning refers to algorithms that target a value function and its derivatives as part of the training. In our context, we would gain control over the approximated functions that are learned ensuring the regularity of the energy functionals. It has been recently formulated for a series of works, and its implementation in the sloshing problem could make the problem take a step forward in the development of physics-informed learning for fluids evaluating its performance in reduced order manifolds.

Other formulations have been hypothesized to simplify the learning process. Contact geometry can lay the foundations for an algorithm that learns only one potential. However, entropy S is usually one of the variables employed in the formulation and we cannot measure it. As a solution this alternative formulation could be coupled with the RNN proposed that recovers the information from the observable variables of the dynamical system.

6.4.2 Fluid modeling

The descriptions obtained from the method output accurate simulations of the integration of the dynamics in time. Nevertheless, the fluid problem is bounded to a specific geometry. We could consider different filling levels, which can be evaluated by computer vision methods [Do et al., 2016], and employ different liquid descriptions that adapt to different vessel geometries. An adaptive algorithm of this kind should also be permutation invariant. Fluid particles evolve in time, but the movement was not extremely chaotic and we could employ a fully connected neural network with this approximation provided the exposed hypothesis. Permutation invariant networks would not be affected by the ordering of the particles when processed in the network, and the algorithm would respond properly to changes in the geometry. *Point nets* belong to the field of geometrical deep learning as a generalization of neural networks to different types of geometries to exploit and fulfill their invariances. They would provide the adaptivity required in the case just exposed.

Although these methods have been used for fluid dynamics [Sanchez-Gonzalez et al., 2020], they do not include learning biases combined with the relational or geometric biases already implemented. In addition, its application in a low-dimensional manifold should be explored. The computational cost of training graphs and point nets could be more elevated than in other approaches such as the one presented in this thesis. In addition, we are still constrained by the real-time prerequisites for digital twins and augmented reality connections. Hence, MOR is a step to be taken in graph learning for this type of application.

Also, we think of more general problems such as contact with other elements —a spoon, for instance— or different actions, like pouring. By offering combinations of actions we lay the foundations for widening the range of applicability of the digital twin.

6.4.3 Virtual interaction and representation

SPNNs were trained to operate in a parallel online phase with the camera, but they were not fully coupled. This new system could provide full online integration and reinforcement learning as proposed in [Moya et al., 2020b].

In addition, we decided to facilitate depth estimation of the glass and liquid by positioning the camera in front of the glass. In a more realistic setting, the camera should be able to move, of the liquid could be placed in other positions in the scene. This proposal would then become a non-rigid structure from motion. Works in the field have been developed in the case of deformable objects [Badias et al., 2021]. However, we still are conditioned by the lack of texture of the liquid and the container. The system developed could be complemented by a method trained for appropriate recognition and tracking of transparent objects and, with this information, perform non-rigid structure from motion.

6.5 Publications and contributions

6.5.1 Journal contributions

1. Moya, B., González, D., Alfaro, I., Chinesta, F., & Cueto, E. (2019). Learning slosh dynamics by means of data. Computational Mechanics, 64(2), 511-523.

2. Moya, B., Alfaro, I., González, D., Chinesta, F., & Cueto, E. (2020). Physically sound, self-learning digital twins for sloshing fluids. PloS one, 15(6), e0234569.

3. Moya, B., Badías, A., Alfaro, I., Chinesta, F., & Cueto, E. (2020). Digital twins that learn and correct themselves. International Journal for Numerical Methods in Engineering.

4. Moya, B., Badías, A., González, D., Chinesta, F., & Cueto, E. (2021). Physics perception in sloshing scenes with guaranteed thermodynamic consistency. arXiv preprint arXiv:2106.13301. Accepted in IEEE Transaction and Pattern Analysis and Machine Inelligence.

5. Moya, B., Badías, A., González, D., Chinesta, F., & Cueto, E. (2022). Physics-informed Reinforcement Learning for Perception and Reasoning about Fluids. arXiv preprint arXiv:2203.05775.

6. Moya B., Pichi F., Hesthaven J., A convolutional graph neural network approach to model order reduction to non-linear parametrized PDEs. In preparation.

6.5.2 Book chapters

1. Chinesta, F., Cueto, E., Grmela, M., Moya, B., Pavelka, M., & Šípka, M. (2020, July). Learning physics from data: a thermodynamic interpretation. In Workshop on Joint Structures and Common Foundations of Statistical Physics, Information Geometry and Inference for Learning (pp. 276-297). Springer, Cham.

6.5.3 Oral communications in conferences and workshops

International

1. Hybrid twins based on physically sound incremental learning. MMLDT-CSET 2021-Oral Communication San Diego, US; ONLINE.

2. Deep learning of fluid dynamics from free surface data for full state reconstruction and correction. Eccomas Young Investigators Conference-Oral Communication Valencia, Spain; ONLINE.

3. Physically sound deep learning development of digital twins from partial measurements of real-world data. Coupled Problems in Engineering 2021-Oral Communication Chia Laguna, Italy; ONLINE.

4. Thermodynamics-based learning of fluid dynamics from partial information. Joint European Thermodynamics Conference 2021-Oral Communication Prague, Czech Republic; ONLINE.

5. Hybrid twins for fluid applications. World Congress in Computational Mechanics - Oral Communication Paris, France; ONLINE.

6. Manifold Learning of complex fluid behavior for real-time simulation. Eccomas Young Investigators Conference-Oral Communication Krakow, Poland.

7. Data-driven learning of slosh dynamics. Congress on Numerical Methods in Engineering-Oral Communication Guimaraes, Portugal.

8. Data-driven, reduced-order modeling and simulation of free-surface flows. Coupled problems in Engineering-Oral Communication Sitges, Spain.
9. Data-based manifold learning of sloshing dynamics. DataBest2019-Oral Communication Paris, France.

National

1. Hybrid twins in the field of intuitive physics. Mechanical Engineering Day-Oral participation Zaragoza, Spain.
2. Learning slosh dynamics by means of data. Mechanical Engineering Day-Oral participation Zaragoza, Spain.
3. Learning slosh dynamics by means of data. Fluid Mechanics Symposiums (University of Zaragoza)-Oral participation Zaragoza, Spain.

6.5.4 Poster communications in conferences and workshops

International

1. Digital twins of fluid dynamics for real-time interaction. C2D3 Virtual Symposium 2020 Cambridge, UK; ONLINE.
2. Data-driven learning of slosh dynamics. Congress on Numerical Methods in Engineering Guimaraes, Portugal.

National

1. Data learning of fluid dynamics for physically informed digital twins. IX Young Investigators Day (Aragon Institute of Engineering Research) Zaragoza, Spain.
2. Aprendizaje automático de dinámica de fluidos mediante modelos de datos. VIII Young Investigators Day (Aragon Institute of Engineering Research) Zaragoza, Spain.

6.5.5 Symposium organization

1. Model reduction and artificial intelligence techniques for surrogate and data-assisted models in computational engineering Eccomas Young Investigators Conference 2021 Valencia, Spain; ONLINE. Co-organized session with Alberto Badías and Matteo Giacomini July 2021.

6.5.6 Outreach communications

Results presented in this dissertation have been publicly published as Youtube videos that can be visualized at

- Self-learning digital twins for sloshing fluids. https://www.youtube.com/watch?v=d1JyhPNkLkU&t=1s
- Physics perception in sloshing scenes. https://www.youtube.com/watch?v=Q1b1VpWRVaQ
- Physics-informed Reinforcement Learning for perception and reasoning about fluids. https://www.youtube.com/watch?v=ikPgZMpsCFk&t=4s

Also, the work has been shown to a wider audience in seminars, radio interviews, and outreach articles published in a local newspaper—see appendix B.

6.5.7 Free software

Any person interested in the presented work can check the databases employed in the following GitHub account under general commons license in the following link https://github.com/beatrizmoya.

- Computational fluids. https://github.com/beatrizmoya/sloshingfluids
- Real fluids. https://github.com/beatrizmoya/RLfluidperception

6.6 Research stay

The training was completed with a research stay in a foreign university to expand the knowledge already acquired and make a contribution to the hosting group. A summary of the work is detailed below.

- Supervisor: Prof. Jan S. Hesthaven.
- École Polytechnique Federal de Lausanne (EPFL).
- Dates: from September 1st, 2021 to February 28th, 2022 (181 days).
- Summary of stage:

  The Chair of Computational Mathematics and Simulation Science at the School of Basic Sciences (MCSS) at EPFL, Lausanne, Switzerland, focuses on the application and evaluation of computational methods for time-dependent and parametrized partial differential equations in high-order problems. This research includes works in model order reduction, discontinuous Galerkin and spectral methods, multiscale problems, fractional differential equations, and the use of machine learning. The latter works combine artificial intelligence with physics knowledge to accelerate and optimize existing techniques. The application problems are rather diverse, mainly related to real applications (electromagnetics, plasma physics, combustion or geoscience). More information can be found at https://www.epfl.ch/labs/mcss/.

The research proposal was related to the study and development of techniques for advanced reduced-order modeling, with the combination with techniques from machine learning. Finally, works have been oriented towards model order reduction of complex problems phrased in unstructured domains with machine learning techniques that adapt to the geometrical conditions of the problem. This has been a collaboration with a current postdoc at MCSS, Federico Pichi, to work on bifurcation problems. The main areas of activity have included

- Study of current state of art methods on Convolutional Neural Networks, and their application to structured and unstructured descriptions.
- Analysis and application of new methods for unstructured grid.
- Study of geometrical deep learning, and specifically graph neural networks.
- Proposal of a MOR algorithm based on graph neural networks.
- Quantification of the method and application to complex physics and domains.
- Work at a report of the results.

Additionally, the candidate participated in EPFL activities and workshops:

- Digital twins days.
- Swiss Numerics Day
- Participation in group talks, with a final presentation of the candidate’s work.

The collaboration has resulted in several new results, in particular with an emphasis on graph neural networks in the context of reduced-order models. The results of the work are presented in a report to study a possible contribution based on this work. The plan is to continue the collaboration and complete a high-quality publication based on the candidate’s work.

Appendix A

Conclusiones

A.1 Introducción

Este capítulo pretende dar una revisión general sobre los principales resultados obtenidos durante la realización del trabajo, y que han llevado a sus principales conclusiones. Además, se expondrán posibles de líneas de investigación futuras, así como el plan de diseminación seguido y formación para dar una perspectiva general del trabajo realizado por el candidato a lo largo del proceso de investigación.

A.2 Conclusiones

La tesis propone un marco desde el que plantear métodos para la comprensión del entorno físico observado mediante una secuencia de video. Este problema supone un reto tanto en el campo computacional como para la visión por computador debido a la complejidad de la dinámica de fluidos. El trabajo asienta las bases de métodos basados en datos y guiados por la física para describir la dinámica y aprender estructuras significativas. Estas aportarán una comprensión más profunda de los fenómenos que permita el razonamiento y la toma de desiciones sobre ellos con datos de observaciones reales.

El trabajo explora los pasos del ciclo de percepción y razonamiento. Empezando por el aprendizaje de una aproximación de la dinámica en la que estamos interesados, el método ha evolucionado progresivamente hacia el desarrollo del gemelo digital cognitivo de un líquido para dar interpretación de la dinámica de fenómenos reales con poco datos y medidas parciales. Finalmente, el gemelo digital se ha convertido en un gemelo híbrido, un sistema capaz de corregirse y adaptarse a los nuevos comportamiento percibidos basado en su experiencia y con datos limitados. Merece la pena destacar que este trabajo se ha concebido y visionado desde el campo del modelado computacional y la ingeniería mecánica, y que pretende proponer una solución que combine métodos de diferentes campos para crear una solución global al problema de percepción propuesto.

El aprendizaje profundo aplicado a la simulación conlleva afrontar varios retos. Además, se desea obtener soluciones generales que a su vez aseguren la consistencia de los resultados, pero hay una gran dependencia en los datos que se utilizan en estas aplicaciones. La preparación de los datos puede ayudar en el proceso de entrenamiento, y una de las posibles técnicas a aplicar es la reducción de modelos. Los métodos de reducción de orden (MOR) encuentran representaciones de los datos en espacios menor dimensionalidad no sólo para acelerar el tiempo de entrenamiento y cálculo sino en la eficiencia del entrenamiento, teniendo un gran impacto en el de desarrollo de métodos basados en datos, como en aplicaciones de regresión y clasificación.

A lo largo de este trabajo se han explorado varias técnicas de aprendizaje automático para cubrir un amplio rango de aplicaciones a lo largo de las distintas etapas que entraña la percepción, que incluye la simulación y predicción, la adquisición de datos y su interpretación, y el razonamiento y corrección.

El proyecto gira en torno a métodos de aprendizaje supervisado para probar la eficacia del uso del formalismo GENERIC como sesgo inductivo en el aprendizaje de modelos. En primer lugar, este enfoque garantiza el cumplimiento de las leyes de la termodinámica, asegurando la conservación de la energía y la disipación de la entropía. Esta característica hace que el algoritmo sea aplicable a dinámicas complejas con grados de libertad no resueltos y con inherente naturaleza disipativa. Esta característica es especialmente importante en casos en los que se simulan fenómenos reales, para los cuales no podemos confiar en aproximaciones *perfectas* sin disipación. En segundo lugar, GENERIC se puede discretizar y aprender con datos, y se puede implementar desde enfoques de aprendizaje distintos. Esto se ha comprobado en el uso GENERIC en dos enfoques diferentes; en el aprendizaje de la variedad de la dinámica, y en su aplicación en redes neuronales.

En tercer lugar, GENERIC muestra una respuesta satisfactoria combinado con métodos de reducción de modelos. Finalmente, la discretización por diferencias finitas no está fuertemente condicionada por el incremento temporal si dicha discretización permite se describe correctamente la dinámica.

Las redes dotan al algoritmo de un mayor grado de generalización ya que estas aprenden una correlación para cualquier nueva entrada del algoritmo. Esta característica es especialmente conveniente para la corrección y adaptación a nuevos comportamientos.

Los métodos de reducción de orden, considerados parte del campo del aprendizaje no supervisado, fueron la clave en el desarrollo de los simuladores físicos. Se ha empleado una descripción del fluido en partículas para su simulación, que se ha complementado con el uso de los métodos MOR para al aprendizaje de las estructuras dinámicas que se encontraban en los datos, alcanzando una comprensión mayor del fenómeno. Los resultados apoyan el uso de métodos no lineales que se adapten a las características de la dinámica de la oscilación de los fluidos. POD requiere un mayor número de modos para describir una evolución, mientras que LLE, TDA, k-PCA, y autoencoders requieren un menor número de dimensiones para aprender satisfactoriamente la estructura GENERIC.

Se han empleado *autoencoders* en los últimos pasos de la implementación de este proyecto por dos motivos; su rapidez de cálculo y las ventajas que ofrece para la fase de corrección. Su comportamiento es bueno, similar al obtenido con TDA y k-PCA. El método TDA, aunque preciso, consumía un alto grado de recursos computacionales, más que los otros dos métodos. Además, el autoencoder se aplica sobre grupos de variables de estado (posición, velocidad, energía y tensiones), capturando más información de cada grupo. El argumento final a favor de su aplicación es el tiempo de simulación. El primer método propuesto, en el segundo capítulo, satisface las restricciones de cálculo en tiempo real realizando una simulación de 1.7 segundos en 1.6404 segundos. Sin embargo, las redes neuronales realizan una simulación de 12 segundos en 3.42 segundos, mejorando la ejecución respecto al método anterior.

Las técnicas MOR también se han explotado en el aprendizaje de variedades del conjunto de líquidos empleados en este trabajo. El método k-PCA aprende un espacio reducido de tres dimensiones en el que se aprecia un agrupamiento de líquidos según sus propiedades y estado dinámico, conveniente para el desarrollo de otras aplicaciones. En este caso, los resultados sirvieron para aprender un clasificador de tipo árbol de decisión que clasificaba el líquido según los datos de entrada.

La adquisición de datos establece una conexión entre el gemelo digital y el líquido real. El algoritmo propuesto obtiene datos satisfactoriamente tanto del vaso real como del líquido. Se ha realizado un ajuste de los parámetros de la cámara para mejorar la precisión de la adquisición de datos usados no sólo en la integración temporal de la dinámica, sino también la corrección.

Hoy en día hay una incuestionable disponibilidad de datos. Sin embargo, no siempre se tiene acceso a los datos necesarios para las descripciones propuestas. En el caso de esta tesis, la cámara sólo tiene acceso a datos de la superficie libre, pero no a variables internas como la energía, que son necesarias para realizar una descripción basada en GENERIC. El aprendizaje semi-supervisado propone un marco para la concepción de ideas para la reconstrucción de la información. Para ello se propone un método que, a partir del análisis de la evolución temporal de la superficie libre, reconstruya el estado dinámico del líquido.

Se ha aplicado aprendizaje por refuerzo en el último paso de desarrollo del gemelo digital cognitivo para alcanzar un mayor grado de adaptación y razonamiento. Al igual que se aprenden acciones por refuerzo en robótica, el razonamiento también se puede entrenar de esta manera, premiando acciones de nuestro aprendizaje que predigan correctamente la evolución dinámica del líquido.

En este trabajo se afrontan dos retos. En primer lugar, el modelo fuente debe haber aprendido previamente unas características suficientemente generales sobre la dinámica para realizar la corrección. Si no, la optimización será dificultosa e ineficiente. En este caso, se ha probado que el uso de técnicas de reducción de orden y el uso de sesgos inductivos benefician la transición en el contexto del aprendizaje por refuerzo. El gemelo digital cognitivo se ha desarrollado respecto a un líquido conocido: la glicerina. Se propone la transferencia de este conocimiento para el aprendizaje de nuevos líquidos, condicionado por las características ya aprendidas en el autoencoder y el entrenamiento del integrador. Los resultados apoyan la coherencia y robustez de esta teoría para una transición suave que reproduzca con exactitud nuevos líquidos percibidos.

En segundo lugar, el aprendizaje por refuerzo no es suficiente por sí mismo para garantizar la optimización. La evaluación que condiciona la recompensa en el aprendizaje por refuerzo debe incluir todo lo que se quiere aprender, en nuestro caso la evolución de la superficie libre y la física. Aunque la recompensa está centrada en la reconstrucción de la superficie libre se propone la inclusión de un término adicional para mantener la consistencia termodinámica de la descripción. De esta manera, la estructura del algoritmo de aprendizaje por refuerzo se vuelve relevante en la optimización para la búsqueda de una solución óptima.

Como resultado se obtiene un gemelo híbrido cognitivo que aprende nuevos comportamientos de fluidos con secuencias de video cortas, y que es capaz de reproducir simulaciones con datos nuevos no percibidos previamente. Además, se consigue reconstruir una aproximación del campo solución completo (velocidad, energía y tensiones) para cualquier nuevo fluido que se manipule.

A.3 Contribuciones de la tesis

Este trabajo ha contribuido al desarrollo del un sistema completo y operativo de percepción capaz de predecir la dinámica de la oscilación en tiempo real, e imitando el comportamiento de fluidos nuevos, con sesgos inductivos. Además, la simulación se complementa con la reconstrucción de los campos solución con observaciones parciales de la superficie libre. Este trabajo se realizado a través de la realización de las siguientes contribuciones:

- Exploración de técnicas de manifold learning aplicadas a la dinámica de la salpicadura para problemas de regresión. Se ha comparado la técnica POD con LLE, TDA k-PCA y autoencoders, probando la eficacia de las técnicas no lineales para capturas las características de la dinámica.

- Desarrollo de un método de reducción de modelo basado en TDA. Este método ha sido implementado para proponer un enfoque que preservase la topología del sistema completo en el espacio de orden reducido.

- Aplicación de técnicas de aprendizaje automático basadas en el formalismo GENERIC para el aprendizaje por regresión de la dinámica de fluidos. Este formalismo ha sido empleado en dos enfoques: *manifold learning*, y redes neuronales.

- Preparación de un sistema de seguimiento para el vaso del líquido acoplado a un algoritmo de simulación.

- Implementación de un gemelo digital y un sistema de visualización de la solución mediante realidad aumentada para dar una interpretación de los resultados y conectar mundos virtual y real.
- Estudio de espacios de orden reducido para el conjunto de fluidos. Exploración y análisis del agrupamiento según sus propiedades, y entrenamiento de un clasificador para distinguir fluidos.
- Preparación de un sistema de adquisición de datos, estimación de profundidad, seguimiento y detección de la superficie libre.
- Desarrollo de una red recurrente para conectar medidas parciales con el espacio reducido de la dinámica de la salpicadura. Dada esa proyección, la solución completa puede ser reconstruida en el momento actual o en pasos de tiempo posteriores.
- Desarrollo de un algoritmo de aprendizaje por refuerzo para la corrección y adaptación a fluidos nuevos no vistos previamente por el sistema con información de la superficie libre. Reconstrucción posterior de la aproximación de la solución completa (velocidad, energía y tensiones) para los nuevos fluidos simulados en tiempo.

A.4 Líneas de trabajo futuras

El presente trabajo propone avances en el campo de la percepción y razonamiento de fluidos. Sin embargo, la aparición de nuevas teorías de aprendizaje permite la evolución en el desarrollo de estos métodos. Las propuestas se pueden agrupar en los tres campos en los que se divide la tesis.

A.4.1 GENERIC en el aprendizaje

La implementación de GENERIC como sesgo inductivo ha suscitado curiosidad en la comunidad debido a las características expuestas previamente. No obstante, su implementación puede verse desde distintas perspectivas, proponiendo diferentes formulaciones para las condiciones de degeneración. En este trabajo se ha considerado la aplicación de las condiciones de degeneración como una restricción suave en el aprendizaje, asumiendo un error numérico bajo en el entrenamiento.

El llamado *aprendizaje Sobolev* se refiere a aquellos algoritmos que entrenan sobre la función objetivo y sus derivadas. En nuestro contexto, este tipo de aprendizaje permitiría ganar control sobre las funciones aprendidas, así como asegurar la regularidad de los funcionales de energía. Se ha aplicado recientemente en trabajos de este campo, y su implementación en el problema de la oscilación de fluidos podría suponer avances en el desarrollo de métodos de aprendizaje guiados por la física, evaluando su comportamiento en espacios de orden reducido de dichos fenómenos.

Se ha propuesto el uso de diferentes formulaciones para simplificar el proceso de aprendizaje. La geometría de contacto podría establecer las bases para un algoritmo que aprenda únicamente un potencial. No obstante, la entropía S es parte de la formulación de variables de este potencial, y no puede ser medido. Como solución, esta formulación alternativa se podría complementar con una red recurrente que recupere información no observable mediante el análisis de la historia de los sistemas con las variables observables.

A.4.2 Modelado de fluidos

Las descripciones obtenidas con el método propuesto proporcionan simulaciones precisas de la dinámica. Sin embargo, se podrían considerar diferentes niveles de llenado, que serían detectados mediante técnicas de visión por computador [Do et al., 2016], y emplear distintas descripciones que se adaptaran a geometrías de vaso diferentes para hacer este método más general.

Un algoritmo adaptativo de estas características debería ser invariante a la permutación. Las partículas de la discretización del fluido evolucionan en el tiempo, pero su movimiento no es extremadamente caótico y podemos emplear una red simple con esta hipótesis. Una formulación invariante a la permutación no se vería afectada por el orden de las partículas cuando las procese la red, y el algoritmo respondería a cambios en la geometría. *Point nets* pertenecen al campo del aprendizaje profundo geométrico como generalización de las redes neuronales para adaptarse a diferentes geométricas y explotar y cumplir con sus invarianzas. Estos métodos dotarían de adaptabilidad necesaria en el caso expuesto.

A pesar de ser ampliamente utilizados en dinámica de fluidos, estos métodos empiezan a incluir sesgos de aprendizaje y física de manera incipiente. Además, su aplicación en espacios reducidos aún no se ha explorado en profundidad. El coste computacional de entrenamiento de grafos y *point nets* podría ser más elevado que en otros métodos, como el presentado en esta tesis. También cabe destacar que en este caso el aprendizaje sigue limitado por las condiciones de respuesta en tiempo real. Por lo tanto, se emplearían técnicas MOR como paso necesario en el aprendizaje de grafos tanto para reducir el coste computacional en el entrenamiento como la realizar la simulación.

El problema se ha limitado a una única dirección de oscilación, y un siguiente paso consistiría en la ampliación de rangos de movimiento. También se han planteado nuevos casos en los que hay interacción fluido-sólido, como el uso de una cuchara, u otras acciones, como el vertido de líquidos. La combinación de pequeñas habilidades puede asentar las bases para la ampliación del rango de aplicaciones de los gemelos digitales.

A.4.3 Interacción virtual y representación

Las redes SPNNs se han entrenado para operar en paralelo en una fase online con la cámara, pero no están completamente acopladas.

El desarrollo del sistema totalmente conectado supondría la integración completa de la simulación y el proceso de aprendizaje propuesto en [Moya et al., 2020b].

Además, se decide facilitar la estimación de la profundidad del vaso y el líquido fijando la cámara enfrente de ellos. En un contexto más realista, la cámara debe poder moverse, y el líquido puede encontrarse en cualquier punto de la escena. Entonces, esta propuesta se comportaría como un problema *non-rigid structure from motion*. Se han desarrollado trabajos en este campo para casos de objetos deformables [Badias et al., 2021]. Sin embargo, seguimos condicionados por la falta de textura de los líquidos y el vaso. El sistema a desarrollar podría ser complementado por un método entrenado para el reconocimiento y detección de este tipo de objetos y, con esta información, se desarrollaría la aplicación propuesta.

A.5 Publicaciones y contribuciones

A.5.1 Publicaciones en revista científica

1. Moya, B., González, D., Alfaro, I., Chinesta, F., & Cueto, E. (2019). Learning slosh dynamics by means of data. Computational Mechanics, 64(2), 511-523.

2. Moya, B., Alfaro, I., González, D., Chinesta, F., & Cueto, E. (2020). Physically sound, self-learning digital twins for sloshing fluids. PloS one, 15(6), e0234569.

3. Moya, B., Badías, A., Alfaro, I., Chinesta, F., & Cueto, E. (2020). Digital twins that learn and correct themselves. International Journal for Numerical Methods in Engineering.

4. Moya, B., Badías, A., González, D., Chinesta, F., & Cueto, E. (2021). Physics perception in sloshing scenes with guaranteed thermodynamic consistency. arXiv preprint arXiv:2106.13301. Accepted in IEEE Transaction and Pattern Analysis

5. Moya, B., Badías, A., González, D., Chinesta, F., & Cueto, E. (2022). Physics-informed Reinforcement Learning for Perception and Reasoning about Fluids. arXiv preprint arXiv:2203.05775.

6. Moya B., Pichi F., Hesthaven J., A convolutional graph neural network approach to model order reduction to non-linear parametrized PDEs. En preparación.

A.5.2 Capítulos de libro

1. Chinesta, F., Cueto, E., Grmela, M., Moya, B., Pavelka, M., & Šípka, M. (2020, July). Learning physics from data: a thermodynamic interpretation. In Workshop on Joint Structures and Common Foundations of Statistical Physics, Information Geometry and Inference for Learning (pp. 276-297). Springer, Cham.

A.5.3 Participación en congresos

Internacionales

1. Hybrid twins based on physically sound incremental learning. MMLDT-CSET 2021- San Diego, EE.UU.; ONLINE.
2. Deep learning of fluid dynamics from free surface data for full state reconstruction and correction. Eccomas Young Investigators Conference- Valencia, España; ONLINE.
3. Physically sound deep learning development of digital twins from partial measurements of real-world data. Coupled Problems in Engineering 2021- Chia Laguna, Italia; ONLINE.
4. Thermodynamics-based learning of fluid dynamics from partial information. Joint European Thermodynamics Conference 2021-Praga, República Checa; ONLINE.
5. Hybrid twins for fluid applications. World Congress in Computational Mechanics - París, Francia; ONLINE.
6. Manifold Learning of complex fluid behavior for real-time simulation. Eccomas Young Investigators Conference- Cracovia, Polonia.
7. Data-driven learning of slosh dynamics. Congress on Numerical Methods in Engineering- Guimaraes, Portugal.
8. Data-driven, reduced-order modeling and simulation of free-surface flows. Coupled problems in Engineering- Sitges, España.
9. Data-based manifold learning of sloshing dynamics. DataBest2019- Paris, Francia.

Nacionales

1. Hybrid twins in the field of intuitive physics. Jornadas de Ingeniería Mecánica 2021 Zaragoza, España.
2. Learning slosh dynamics by means of data. Jornadas de Ingeniería mecánica 2019 Zaragoza, España.
3. Learning slosh dynamics by means of data. Seminarios de Mecánica de Fluidos(Universidad de Zaragoza)-Zaragoza, España.

A.5.4 Participaciones con póster en conferencias y workshops

Internacionales

1. Digital twins of fluid dynamics for real-time interaction. C2D3 Virtual Symposium 2020 Cambridge, Inglaterra; ONLINE.

2. Data-driven learning of slosh dynamics. Congress on Numerical Methods in Engineering Guimaraes, Portugal.

Nacionales

1. Data learning of fluid dynamics for physically informed digital twins. IX Young Investigators Day (Aragon Institute of Engineering Research) Zaragoza, España.
2. Aprendizaje automático de dinámica de fluidos mediante modelos de datos. VIII Young Investigators Day (Aragon Institute of Engineering Research) Zaragoza, España.

A.5.5 Organización de sesiones en congresos

1. Model reduction and artificial intelligence techniques for surrogate and data-assisted models in computational engineering Eccomas Young Investigators Conference 2021 Valencia, España; ONLINE. Sesión coorganizada con Alberto Badías y Matteo Giacomini en julio de 2021.

A.5.6 Diseminación general

Los resultados de esta tesis se han mostrado públicamente como vídeos en la plataforma Youtube:

- Self-learning digital twins for sloshing fluids. https://www.youtube.com/watch?v=d1JyhPNkLkU&t=1s
- Physics perception in sloshing scenes. https://www.youtube.com/watch?v=Q1b1VpWRVaQ
- Physics-informed Reinforcement Learning for perception and reasoning about fluids. https://www.youtube.com/watch?v=ikPgZMpsCFk&t=4s

Además, el trabajo ha sido diseminado para una audiencia más amplia en seminarios, entrevistas de radio, y artículos publicados en prensa local (ver anexo B).

A.5.7 Software libre

Cualquier persona interesada puede consultar las bases de datos empleadas en la siguiente cuenta de GitHub bajo licencia *general commons* en el siguiente enlace https://github.com/beatrizmoya.

- Fluidos computacionales. https://github.com/beatrizmoya/sloshingfluids
- Fluidos reales. https://github.com/beatrizmoya/RLfluidperception

A.6 Estancias de doctorado

La formación se ha complementado con una estancia de doctorado en un centro extranjero para expandir los conocimientos adquiridos, así como hacer una contribución al grupo receptor. A continuación se detalla un resumen del trabajo realizado.

- Supervisor: Dr. Jan S. Hesthaven.
- École Polytechnique Federal de Lausanne (EPFL).
- Fecha: desde el 1 de septiembre de 2021 hasta el 28 de febrero de 2022 (181 días).
- Resumen de la estancia:

  El departamento de matemática computacional y simulación de la escuela de ciencias básicas (MCSS) en la universidad EPFL, Lausana, Suiza, centra su actividad en la aplicación y evaluación de métodos computacionales para problemas dependientes del tiempo y parametrizados en derivadas parciales de alto orden. Esta investigación incluye trabajos en métodos de reducción de orden, Galerkin discontinuo y métodos espectrales, problemas multiescalada, ecuaciones diferenciales fraccionadas, y el uso de aprendizaje automático. El último se combina con inteligencia artificial y conocimiento de la física para acelerar y optimizar técnicas existentes. La aplicación a problemas es diversa, principalmente relacionada con aplicaciones reales (electromagnetismo, física de plasma, combustión o geociencia). Puede encontrarse más información en `https://www.epfl.ch/labs/mcss/`

  La propuesta de investigación estaba relacionada con el estudio y desarrollo de técnicas avanzadas de reducción de modelos combinadas con técnicas de machine learning. Finalmente, el trabajo se orientó a la reducción de modelos en físicas complejas expresados en mallas no estructuradas con técnicas de machine learning que se adaptan a las condiciones geométricas del problema. Este trabajo ha sido una colaboración con un trabajador en un post-doctorado en el grupo, Federico Pichi, para trabajar en problemas de bifurcación. Las tareas principales han sido:

  - Estudio del estado del arte con énfasis en redes convolucionales, y su aplicación en problemas cartesianos.
  - Análisis y aplicación de métodos nuevos en mallas desestructuradas.
  - Estudio de métodos de aprendizaje profundo con sesgos geométricos, específicamente redes de grafos.
  - Propuesta de una técnica de reducción basada en grafos.
  - Evaluación del método en dominios físicos complejos.
  - Preparación de informed de resultados.

Además, el candidato ha participado en varias actividades y workshops en la universidad EPFL:

- Digital twins days.
- Swiss Numerics Day.
- Participation en charlas del departamento, con una presentación final del trabajo del candidato.

La colaboración produjo varios resultados, en particular haciendo hincapié en las redes de grafos en el contexto de la reducción de modelos. Los resultados de este trabajo son presentados en un resumen para estudiar una posible contribución a revista basada en dichos resultados.

Appendix B

Artículo de divulgación: Aprender a razonar, el ser o no ser de la robótica

- Aprender a razonar, el ser o no ser de la robótica. Heraldo de Aragón, suplemento Tercer Milenio. 20 noviembre 2021.

Los niños aprenden cómo funciona el mundo observando, experimentando y jugando. Así desarrollan la intuición y el razonamiento. ¿Podría llegar a hacer esto un robot? Las máquinas autónomas son uno de los grandes desafíos científicos del siglo XXI. Para su desarrollo es necesario diseñar sistemas que les ayuden a percibir y entender el mundo que les rodea de una manera parecida a la humana. Parte de la estrategia en inteligencia artificial de la Universidad de Zaragoza orienta su trabajo hacia el desarrollo de estos sistemas.

Ver, razonar y aprender son tres habilidades inherentes de las personas. El cerebro humano procesa cada imagen captada por los ojos en 13 milésimas de segundo y con ellas hace una foto de la realidad y sus leyes físicas. Este mecanismo nos permite aprender cuando somos niños, y se mantiene activo en la edad adulta: "¿Se caerá la torre de bloques? ¿Qué ocurre si lanzo una pelota?".

Para hacer estas predicciones, nuestro cerebro realiza rápidas simulaciones basadas en la información que ha aprendido y los principios físicos que ha deducido. El desarrollo de esta intuición es también de gran interés en la robótica. El objetivo de estos trabajos es crear robots que sean independientes y aprendan a razonar sobre el mundo que los rodea.

Para que los robots interactúen con el entorno no solo queremos controlar sus propios movimientos, sino también las consecuencias de sus actos, y eso se consigue evaluándolas en tiempo real con una simulación del mundo que imite sus sentidos y la percepción de lo que ocurre a su alrededor. Desde el laboratorio en Inteligencia Artificial del Instituto de Investigación en Ingeniería de Aragón se desarrolla una investigación para convertir los valores de los píxeles de las imágenes en información que entienda un ordenador. El objetivo es predecir el comportamiento de líquidos, muy presentes en tareas de cuidado y procesos industriales, para poder tomar decisiones informadas. ¿Pero qué distingue esta tecnología de otras?

Los trabajos existentes tienen dos problemas: la carencia de física, que puede resultar en situaciones incongruentes, y sus largos tiempos de cómputo para tomar decisiones. Esta investigación afronta esos retos para dar una solución que, además de estar dentro de los límites de la física, pueda dar una respuesta en tiempo real.

¿Qué son los gemelos digitales?

Para que un robot comprenda el mundo es necesario simularlo, es decir, crear una copia virtual para traducirlo al lenguaje de un ordenador. Los gemelos digitales son copias de máquinas, productos o servicios que se comportan de manera idéntica a su parte real. Con ellos aprendemos la respuesta de un sistema sin necesidad de alterar la copia original. Por ejemplo, podemos probar la acción de un medicamento en el gemelo digital de un corazón humano para comprobar si va a ser efectivo. Entre los gemelos digitales, distinguimos los gemelos vivos, que son aquellos que reaccionan a la vez o incluso más rápido que la parte real. Esta tecnología permite adelantarse a posibles problemas y dar soluciones preventivas. Diseñando gemelos vivos de líquidos, un robot interpreta lo que está manipulando.

¿Cuál es el próximo reto de los robots inteligentes?

Lejos de ser sustitutos, los robots independientes se crean para asistir a las personas en tareas que van desde lo cotidiano hasta lo peligroso, pero aún quedan grandes retos a los que hacer frente en su desarrollo. A pesar de tener acceso a grandes cantidades de datos, en ocasiones los que necesitamos son inaccesibles o difíciles de conseguir. Por tanto, es necesario crear sistemas que recuperen la información importante y que no sea medible para avanzar en el desarrollo de aplicaciones.

A su vez, aunque los modelos de percepción existentes pueden ser precisos, pueden darse situaciones fuera de su rango de aprendizaje. Los sistemas inteligentes deben evolucionar hacia modelos adaptativos capaces de detectar que el modelo no se ajusta completamente a la realidad que percibe y corregirse. Esto es lo que se llama un gemelo híbrido.

¿Hasta dónde llega la ciencia de datos?

En la era del internet de las cosas existen millones de datos que podemos aprovechar, pero también es necesario aprender a trabajar con este volumen de información: ¿Qué información dan los datos? ¿Qué conocimiento esconden?

Un modelo es la réplica matemática de un fenómeno real con el cual queremos trabajar. Los modelos aprendidos a partir de datos, en nuestro caso imágenes, son cada vez más comunes y se han visto fuertemente influenciados por las últimas tendencias en inteligencia artificial.

En particular, las redes neuronales son estructuras que imitan las conexiones cerebrales en el proceso de aprendizaje. A pesar de su poder, existen inconvenientes en su uso. Las

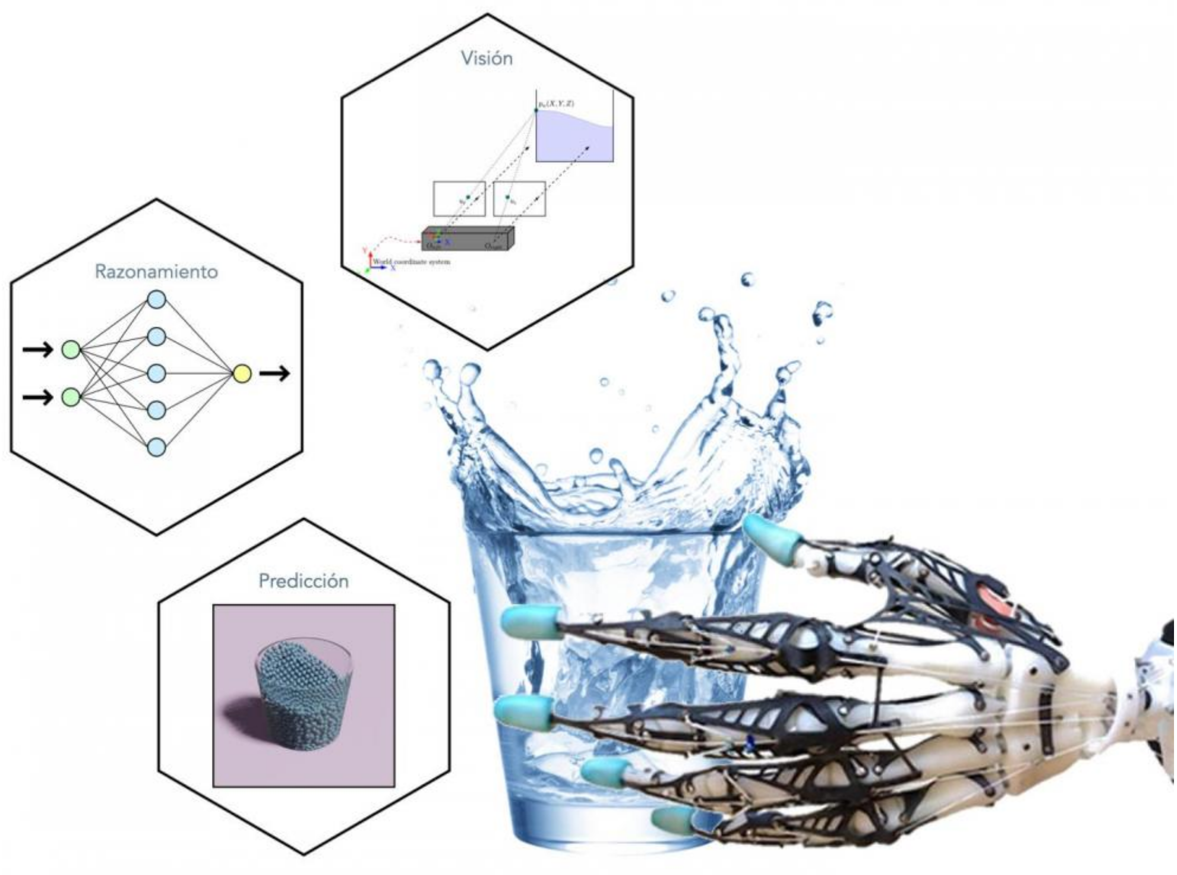


Figure B.1: Representación del ciclo cognitivo para la interpretación de dinámica de fluidos

redes encuentran una solución buena con la información de la que disponen, pero puede no ser la más óptima. Incorporar conocimientos de la física adquiridos durante siglos de avances científicos es la clave para guiar el aprendizaje de aquello que intentamos modelar. Pero trabajar con esta cantidad de datos podría ser computacionalmente inviable. Entre las técnicas matemáticas que forman la ciencia de datos se encuentran las llamadas técnicas de reducción de modelos, que analizan la información real que dan los datos para reducir su complejidad y aprender un sistema más sencillo pero con la misma información. Así podemos lograr tiempos de respuesta en tiempo real.

Publicado en https://www.heraldo.es/noticias/aragon/2021/11/20/aprender-a-razonar-el-ser-o-no-ser-de-la-robotica-1534983.html
Published in https://www.heraldo.es/noticias/aragon/2021/11/20/aprender-a-razonar-el-ser-o-no-ser-de-la-robotica-1534983.html

Appendix C

Figments of reality

The following picture, named *Figments of reality*, was finalist in the Arts and Science contest in the World Congress in Computational Mechanics 2020:

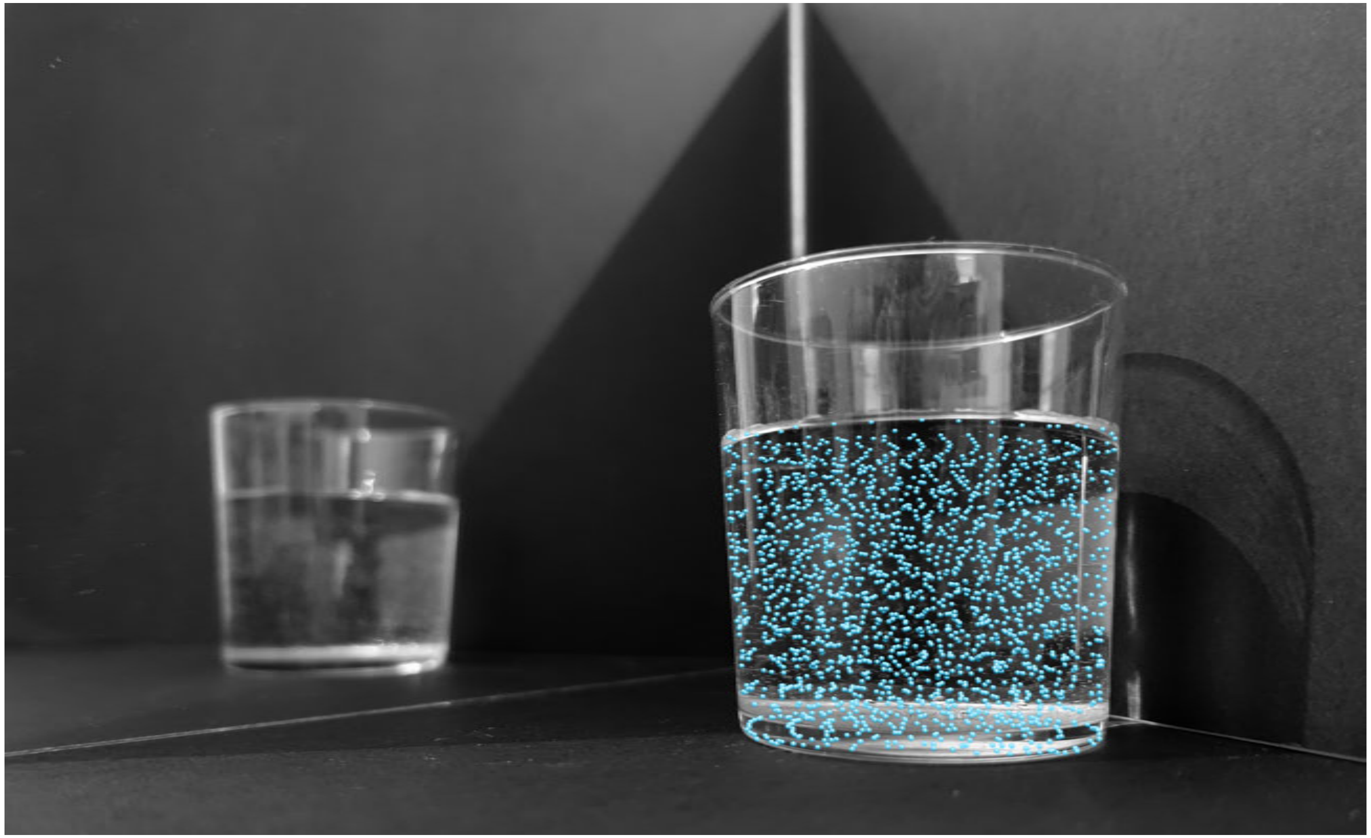


Figure C.1: The research revolves around the development of living digital twins of fluid dynamics. The avatar of the fluid, discretized in particles, learns from reality and shows the results of the simulation with augmented reality, as we see in the picture. These twins are a reflection of a real fluid, like the reflection of the mirror. Even though they are not real, they act as if they were. Their behavior is portrayed in the picture by representing the particles as a recreation of the real fluid, and not the one mirrored. We don't have to find out the reality, because for machines this is the reality.

Appendix D

Abbreviations

- AE: autoencoder
- AI: Artificial Intelligence
- AGI: artificial general intelligence
- ANN: Artificial Neural Network
- AR: Augmented Reality
- AV: Augmented Virtuality
- CNN: Convolutional Neural Network
- DMD: Dynamic mode decomposition
- EIM: Empirical Interpolation Method
- fps: frames per second
- GPU: Graphics Processing Unit
- IA: Intelligence Augmentation
- k-PCA: kernel Principal Component Analysis
- k-PGD: kernel Principal Generalized Decomposition
- LDA: Linear Discriminant Analysis
- ML: Machine Learning
- MOR: Model Order Reduction
- MR: Mixed Reality
- NRSfM: Non-rigid Structure from Motion
- PCA: Principal Component Analysis

- PDE: Partial Derivative Equation
- PGD: Proper Generalized Decomposition
- PINNs: Physically Informed Neural Networks
- RB: Reduced Basis
- RL: reinforcement learning
- RMSE: Root Mean Square Error
- RNN: Recurrent Neural Network
- RR: Real Reality
- rs-PGD: regularized Proper Generalized Decomposition
- SINDy: Sparse identification of non-linear dynamics
- s-PGD: sparse Proper Generalized Decomposition
- s^2-PGD: doubly sparse Proper Generalized Decomposition
- SPNN: Structure-Preserving Neural Networks
- SfM: Structure from Motion
- SLAM: Simultaneous Localization And Mapping
- SVD: Singular Value Decomposition
- VR: Virtual Reality

Bibliography

Bibliography

[Ahmed et al., 2021] Ahmed, S. E., San, O., Rasheed, A., and Iliescu, T. (2021). Nonlinear proper orthogonal decomposition for convection-dominated flows. *Physics of Fluids*, 33(12):121702.

[Alla and Kutz, 2017] Alla, A. and Kutz, J. N. (2017). Nonlinear model order reduction via dynamic mode decomposition. *SIAM Journal on Scientific Computing*, 39(5):B778–B796.

[Allen et al., 2020] Allen, K. R., Smith, K. A., and Tenenbaum, J. B. (2020). Rapid trial-and-error learning with simulation supports flexible tool use and physical reasoning. *Proceedings of the National Academy of Sciences*, 117(47):29302–29310.

[Atkeson and Schaal, 1997] Atkeson, C. G. and Schaal, S. (1997). Learning tasks from a single demonstration. In *Proceedings of International Conference on Robotics and Automation*, volume 2, pages 1706–1712. IEEE.

[Aversano et al., 2019] Aversano, G., Bellemans, A., Li, Z., Coussement, A., Gicquel, O., and Parente, A. (2019). Application of reduced-order models based on pca & kriging for the development of digital twins of reacting flow applications. *Computers & chemical engineering*, 121:422–441.

[Ayensa-Jiménez et al., 2020] Ayensa-Jiménez, J., Doweidar, M. H., Sanz-Herrera, J. A., and Doblaré, M. (2020). On the application of physically-guided neural networks with internal variables to continuum problems. *arXiv preprint arXiv:2011.11376*.

[Ayensa-Jiménez et al., 2021] Ayensa-Jiménez, J., Doweidar, M. H., Sanz-Herrera, J. A., and Doblaré, M. (2021). Prediction and identification of physical systems by means of physically-guided neural networks with meaningful internal layers. *Computer Methods in Applied Mechanics and Engineering*, 381:113816.

[Badías et al., 2018] Badías, A., Alfaro, I., González, D., Chinesta, F., and Cueto, E. (2018). Reduced order modeling for physically-based augmented reality. *Computer Methods in Applied Mechanics and Engineering*, 341:53–70.

[Badias et al., 2021] Badias, A., Alfaro, I., Gonzalez, D., Chinesta, F., and Cueto, E. (2021). Morph-dslam: Model order reduction for physics-based deformable slam. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

[Badías et al., 2019] Badías, A., Curtit, S., González, D., Alfaro, I., Chinesta, F., and Cueto, E. (2019). An augmented reality platform for interactive aerodynamic design and analysis. *International Journal for Numerical Methods in Engineering*, 120(1):125–138.

[Badías et al., 2020] Badías, A., González, D., Alfaro, I., Chinesta, F., and Cueto, E. (2020). Real-time interaction of virtual and physical objects in mixed reality applications. *International Journal for Numerical Methods in Engineering*, 121(17):3849–3868.

[Bai et al., 2017] Bai, Z., Brunton, S. L., Brunton, B. W., Kutz, J. N., Kaiser, E., Spohn, A., and Noack, B. R. (2017). Data-driven methods in fluid dynamics: Sparse classification from experimental data. In *Whither Turbulence and Big Data in the 21st Century?*, pages 323–342. Springer.

[Bassenne and Lozano-Durán, 2019] Bassenne, M. and Lozano-Durán, A. (2019). Computational model discovery with reinforcement learning. *arXiv preprint arXiv:2001.00008*.

[Bates et al., 2015] Bates, C., Battaglia, P. W., Yildirim, I., and Tenenbaum, J. B. (2015). Humans predict liquid dynamics using probabilistic simulation. In *CogSci*. Citeseer.

[Battaglia et al., 2016] Battaglia, P., Pascanu, R., Lai, M., Jimenez Rezende, D., et al. (2016). Interaction networks for learning about objects, relations and physics. *Advances in neural information processing systems*, 29.

[Battaglia et al., 2018] Battaglia, P. W., Hamrick, J. B., Bapst, V., Sanchez-Gonzalez, A., Zambaldi, V., Malinowski, M., Tacchetti, A., Raposo, D., Santoro, A., Faulkner, R., et al. (2018). Relational inductive biases, deep learning, and graph networks. *arXiv preprint arXiv:1806.01261*.

[Battaglia et al., 2013] Battaglia, P. W., Hamrick, J. B., and Tenenbaum, J. B. (2013). Simulation as an engine of physical scene understanding. *Proceedings of the National Academy of Sciences*, 110(45):18327–18332.

[Bender and Koschier, 2015] Bender, J. and Koschier, D. (2015). Divergence-free smoothed particle hydrodynamics. In *Proceedings of the 14th ACM SIGGRAPH/Eurographics symposium on computer animation*, pages 147–155.

[Benosman et al., 2021] Benosman, M., Chakrabarty, A., and Borggaard, J. (2021). Reinforcement learning-based model reduction for partial differential equations: Application to the burgers equation. In *Handbook of Reinforcement Learning and Control*, pages 293–317. Springer.

[Bertalan et al., 2019] Bertalan, T., Dietrich, F., Mezić, I., and Kevrekidis, I. G. (2019). On learning hamiltonian systems from data. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 29(12):121107.

[Bieker et al., 2020] Bieker, K., Peitz, S., Brunton, S. L., Kutz, J. N., and Dellnitz, M. (2020). Deep model predictive flow control with limited sensor data and online learning. *Theoretical and Computational Fluid Dynamics*, pages 1–15.

[Bogatskiy et al., 2022] Bogatskiy, A., Ganguly, S., Kipf, T., Kondor, R., Miller, D. W., Murnane, D., Offermann, J. T., Pettee, M., Shanahan, P., Shimmin, C., et al. (2022). Symmetry group equivariant architectures for physics. *arXiv preprint arXiv:2203.06153*.

[Bongard and Lipson, 2007] Bongard, J. and Lipson, H. (2007). Automated reverse engineering of nonlinear dynamical systems. *Proceedings of the National Academy of Sciences*, 104(24):9943–9948.

[Breiman, 2001] Breiman, L. (2001). Random forests. *Machine learning*, 45(1):5–32.

[Bronstein et al., 2021] Bronstein, M. M., Bruna, J., Cohen, T., and Veličković, P. (2021). Geometric deep learning: Grids, groups, graphs, geodesics, and gauges. *arXiv preprint arXiv:2104.13478*.

[Brunton et al., 2020] Brunton, S. L., Noack, B. R., and Koumoutsakos, P. (2020). Machine learning for fluid mechanics. *Annual Review of Fluid Mechanics*, 52:477–508.

[Brunton et al., 2016] Brunton, S. L., Proctor, J. L., and Kutz, J. N. (2016). Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *Proceedings of the national academy of sciences*, 113(15):3932–3937.

[Bucci et al., 2019] Bucci, M. A., Semeraro, O., Allauzen, A., Wisniewski, G., Cordier, L., and Mathelin, L. (2019). Control of chaotic systems by deep reinforcement learning. *Proceedings of the Royal Society A*, 475(2231):20190351.

[Buisson-Fenet et al., 2020] Buisson-Fenet, M., Solowjow, F., and Trimpe, S. (2020). Actively learning gaussian process dynamics. In *Learning for dynamics and control*, pages 5–15. PMLR.

[Bukka et al., 2021] Bukka, S. R., Gupta, R., Magee, A. R., and Jaiman, R. K. (2021). Assessment of unsteady flow predictions using hybrid deep learning based reduced-order models. *Physics of Fluids*, 33(1):013601.

[Callaham et al., 2019] Callaham, J. L., Maeda, K., and Brunton, S. L. (2019). Robust flow reconstruction from limited measurements via sparse representation. *Physical Review Fluids*, 4(10):103907.

[Carrara et al., 2020] Carrara, P., De Lorenzis, L., Stainier, L., and Ortiz, M. (2020). Data-driven fracture mechanics. *Computer Methods in Applied Mechanics and Engineering*, 372:113390.

[Carrara et al., 2021] Carrara, P., Ortiz, M., and De Lorenzis, L. (2021). Data-driven rate-dependent fracture mechanics. *Journal of the Mechanics and Physics of Solids*, 155:104559.

[Carroll and Holt, 1972] Carroll, M. and Holt, A. C. (1972). Suggested modification of the p-α model for porous materials. *Journal of Applied Physics*, 43(2):759–761.

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

[Chen et al., 2021a] Chen, W., Wang, Q., Hesthaven, J. S., and Zhang, C. (2021a). Physics-informed machine learning for reduced-order modeling of nonlinear problems. *Journal of Computational Physics*, 446:110666.

[Chen et al., 2021b] Chen, Y., Hosseini, B., Owhadi, H., and Stuart, A. M. (2021b). Solving and learning nonlinear pdes with gaussian processes. *Journal of Computational Physics*, 447:110668.

[Chinesta et al., 2018] Chinesta, F., Cueto, E., Abisset-Chavanne, E., Duval, J., and Khaldi, F. (2018). Virtual, digital and hybrid twins: A new paradigm in data-based engineering and engineered data. *Archives of Computational Methods in Engineering*.

[Chinesta et al., 2020] Chinesta, F., Cueto, E., Abisset-Chavanne, E., Duval, J. L., and El Khaldi, F. (2020). Virtual, digital and hybrid twins: a new paradigm in data-based engineering and engineered data. *Archives of computational methods in engineering*, 27(1):105–134.

[Cho et al., 2014] Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y. (2014). Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.

[Chung et al., 2014] Chung, J., Gulcehre, C., Cho, K., and Bengio, Y. (2014). Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*.

[Course et al., 2020] Course, K., Evans, T., and Nair, P. (2020). Weak form generalized hamiltonian learning. *Advances in Neural Information Processing Systems*, 33:18716–18726.

[Cranmer et al., 2020a] Cranmer, M., Greydanus, S., Hoyer, S., Battaglia, P., Spergel, D., and Ho, S. (2020a). Lagrangian neural networks. *arXiv preprint arXiv:2003.04630*.

[Cranmer et al., 2020b] Cranmer, M., Sanchez Gonzalez, A., Battaglia, P., Xu, R., Cranmer, K., Spergel, D., and Ho, S. (2020b). Discovering symbolic models from deep learning with inductive biases. *Advances in Neural Information Processing Systems*, 33:17429–17442.

[Cutler et al., 2012] Cutler, A., Cutler, D. R., and Stevens, J. R. (2012). Random forests. In *Ensemble machine learning*, pages 157–175. Springer.

[Dandekar et al., 2020] Dandekar, R., Chung, K., Dixit, V., Tarek, M., Garcia-Valadez, A., Vemula, K. V., and Rackauckas, C. (2020). Bayesian neural ordinary differential equations. *arXiv preprint arXiv:2012.07244*.

[Deisenroth et al., 2013] Deisenroth, M. P., Fox, D., and Rasmussen, C. E. (2013). Gaussian processes for data-efficient learning in robotics and control. *IEEE transactions on pattern analysis and machine intelligence*, 37(2):408–423.

[Do et al., 2016] Do, C., Schubert, T., and Burgard, W. (2016). A probabilistic approach to liquid level detection in cups using an rgb-d camera. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2075–2080. IEEE.

[Drezet and Harrison, 1998] Drezet, P. and Harrison, R. (1998). Support vector machines for system identification.

[Durrant-Whyte and Bailey, 2006] Durrant-Whyte, H. and Bailey, T. (2006). Simultaneous localization and mapping: part i. *IEEE robotics & automation magazine*, 13(2):99–110.

[Eppel, 2016] Eppel, S. (2016). Tracing liquid level and material boundaries in transparent vessels using the graph cut computer vision approach. *arXiv preprint arXiv:1602.00177*.

[Erichson et al., 2020] Erichson, N. B., Mathelin, L., Yao, Z., Brunton, S. L., Mahoney, M. W., and Kutz, J. N. (2020). Shallow neural networks for fluid flow reconstruction with limited sensors. *Proceedings of the Royal Society A*, 476(2238):20200097.

[Erichson et al., 2019] Erichson, N. B., Muehlebach, M., and Mahoney, M. W. (2019). Physics-informed autoencoders for lyapunov-stable fluid flow prediction. *arXiv preprint arXiv:1905.10866*.

[Ershadnia et al., 2020] Ershadnia, R., Amooie, M. A., Shams, R., Hajirezaie, S., Liu, Y., Jamshidi, S., and Soltanian, M. R. (2020). Non-newtonian fluid flow dynamics in rotating annular media: Physics-based and data-driven modeling. *Journal of Petroleum Science and Engineering*, 185:106641.

[Español, 2004] Español, P. (2004). Statistical mechanics of coarse-graining. In *Novel Methods in Soft Matter Simulations*, pages 69–115. Springer.

[Español et al., 1999] Español, P., Serrano, M., and Öttinger, H. C. (1999). Thermodynamically admissible form for discrete hydrodynamics. *Physical review letters*, 83(22):4542.

[Fish, 2005] Fish, D. (2005). *Metriplectic systems*. Portland State University.

[Flaschel et al., 2021] Flaschel, M., Kumar, S., and De Lorenzis, L. (2021). Unsupervised discovery of interpretable hyperelastic constitutive laws. *Computer Methods in Applied Mechanics and Engineering*, 381:113852.

[Fujisawa and Kato, 2009] Fujisawa, M. and Kato, H. (2009). Interactive fluid simulation using augmented reality interface. In *International Conference on Virtual and Mixed Reality*, pages 431–438. Springer.

[Fukami et al., 2021] Fukami, K., Hasegawa, K., Nakamura, T., Morimoto, M., and Fukagata, K. (2021). Model order reduction with neural networks: Application to laminar and turbulent flows. *SN Computer Science*, 2(6):1–16.

[Gao et al., 2021] Gao, H., Sun, L., and Wang, J.-X. (2021). Phygeonet: physics-informed geometry-adaptive convolutional neural networks for solving parameterized steady-state pdes on irregular domain. *Journal of Computational Physics*, 428:110079.

[Garnier et al., 2021] Garnier, P., Viquerat, J., Rabault, J., Larcher, A., Kuhnle, A., and Hachem, E. (2021). A review on deep reinforcement learning for fluid mechanics. *Computers & Fluids*, 225:104973.

[Ghaboussi, 2010] Ghaboussi, J. (2010). Advances in neural networks in computational mechanics and engineering. In *Advances of soft computing in engineering*, pages 191–236. Springer.

[Ghnatios et al., 2019] Ghnatios, C., Alfaro, I., González, D., Chinesta, F., and Cueto, E. (2019). Data-driven generic modeling of poroviscoelastic materials. *Entropy*, 21(12):1165.

[González et al., 2019a] González, D., Chinesta, F., and Cueto, E. (2019a). Learning corrections for hyperelastic models from data.

[González et al., 2019b] González, D., Chinesta, F., and Cueto, E. (2019b). Thermodynamically consistent data-driven computational mechanics. *Continuum Mechanics and Thermodynamics*, 31(1):239–253.

[González et al., 2021] González, D., Chinesta, F., and Cueto, E. (2021). Learning non-markovian physics from data. *Journal of Computational Physics*, 428:109982.

[Goodfellow et al., 2016] Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep Learning*. MIT Press. http://www.deeplearningbook.org.

[Goswami et al., 2020] Goswami, S., Anitescu, C., Chakraborty, S., and Rabczuk, T. (2020). Transfer learning enhanced physics informed neural network for phase-field modeling of fracture. *Theoretical and Applied Fracture Mechanics*, 106:102447.

[Greydanus et al., 2019] Greydanus, S., Dzamba, M., and Yosinski, J. (2019). Hamiltonian neural networks. In *Advances in Neural Information Processing Systems*, pages 15379–15389.

[Grmela and Öttinger, 1997] Grmela, M. and Öttinger, H. C. (1997). Dynamics and thermodynamics of complex fluids. i. development of a general formalism. *Physical Review E*, 56(6):6620.

[Guagliumi et al., 2021] Guagliumi, L., Berti, A., Monti, E., and Carricato, M. (2021). A simple model-based method for sloshing estimation in liquid transfer in automatic machines. *IEEE Access*, 9:129347–129357.

[Guastoni et al., 2021] Guastoni, L., Güemes, A., Ianiro, A., Discetti, S., Schlatter, P., Azizpour, H., and Vinuesa, R. (2021). Convolutional-network models to predict wall-bounded turbulence from wall quantities. *Journal of Fluid Mechanics*, 928.

[Gutierrez-Gomez et al., 2016] Gutierrez-Gomez, D., Mayol-Cuevas, W., and Guerrero, J. J. (2016). Dense rgb-d visual odometry using inverse depth. *Robotics and Autonomous Systems*, 75:571–583.

[Hanuka et al., 2021] Hanuka, A., Huang, X., Shtalenkova, J., Kennedy, D., Edelen, A., Zhang, Z., Lalchand, V., Ratner, D., and Duris, J. (2021). Physics model-informed gaussian process for online optimization of particle accelerators. *Physical Review Accelerators and Beams*, 24(7):072802.

[Hartley and Zisserman, 2003] Hartley, R. and Zisserman, A. (2003). *Multiple view geometry in computer vision*. Cambridge university press.

[He et al., 2015] He, K., Zhang, X., Ren, S., and Sun, J. (2015). Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034.

[Hernández et al., 2022] Hernández, Q., Badías, A., Chinesta, F., and Cueto, E. (2022). Thermodynamics-informed graph neural networks. *arXiv preprint arXiv:2203.01874*.

[Hernández et al., 2021a] Hernández, Q., Badías, A., González, D., Chinesta, F., and Cueto, E. (2021a). Deep learning of thermodynamics-aware reduced-order models from data. *Computer Methods in Applied Mechanics and Engineering*, 379:113763.

[Hernández et al., 2021b] Hernández, Q., Badías, A., González, D., Chinesta, F., and Cueto, E. (2021b). Structure-preserving neural networks. *Journal of Computational Physics*, 426:109950.

[Herrmann, 1969] Herrmann, W. (1969). Constitutive equation for the dynamic compaction of ductile porous materials. *Journal of applied physics*, 40(6):2490–2499.

[Herschel and Bulkley, 1926] Herschel, W. H. and Bulkley, R. (1926). Konsistenzmessungen von gummi-benzollösungen. *Kolloid-Zeitschrift*, 39(4):291–300.

[Hester and Stone, 2012] Hester, T. and Stone, P. (2012). Learning and using models. In *Reinforcement learning*, pages 111–141. Springer.

[Hesthaven et al., 2020] Hesthaven, J. S., Pagliantini, C., and Ripamonti, N. (2020). Rank-adaptive structure-preserving reduced basis methods for hamiltonian systems. *arXiv preprint arXiv:2007.13153*.

[Hesthaven and Ubbiali, 2018] Hesthaven, J. S. and Ubbiali, S. (2018). Non-intrusive reduced order modeling of nonlinear problems using neural networks. *Journal of Computational Physics*, 363:55–78.

[Hinton and Salakhutdinov, 2006] Hinton, G. E. and Salakhutdinov, R. R. (2006). Reducing the dimensionality of data with neural networks. *science*, 313(5786):504–507.

[Hochreiter and Schmidhuber, 1997] Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8):1735–1780.

[Hornik et al., 1989] Hornik, K., Stinchcombe, M., and White, H. (1989). Multilayer feedforward networks are universal approximators. *Neural networks*, 2(5):359–366.

[Huerta and Liu, 1988] Huerta, A. and Liu, W. K. (1988). Viscous flow with large free surface motion. *Computer Methods in Applied Mechanics and Engineering*, 69(3):277–324.

[Huttenlocher et al., 1993] Huttenlocher, D. P., Klanderman, G. A., and Rucklidge, W. J. (1993). Comparing images using the hausdorff distance. *IEEE Transactions on pattern analysis and machine intelligence*, 15(9):850–863.

[Ibáñez et al., 2018a] Ibáñez, R., Abisset-Chavanne, E., Aguado, J. V., González, D., Cueto, E., and Chinesta, F. (2018a). A manifold learning approach to data-driven computational elasticity and inelasticity. *Archives of Computational Methods in Engineering*, 25(1):47–57.

[Ibáñez et al., 2018b] Ibáñez, R., Abisset-Chavanne, E., Ammar, A., González, D., Cueto, E., Huerta, A., Duval, J. L., and Chinesta, F. (2018b). A multidimensional data-driven sparse identification technique: the sparse proper generalized decomposition. *Complexity*, 2018.

[Ibáñez et al., 2019] Ibáñez, R., Abisset-Chavanne, E., González, D., Duval, J.-L., Cueto, E., and Chinesta, F. (2019). Hybrid constitutive modeling: data-driven learning of corrections to plasticity models. *International Journal of Material Forming*, 12(4):717–725.

[Ibáñez et al., 2017] Ibáñez, R., Borzacchiello, D., Aguado, J. V., Abisset-Chavanne, E., Cueto, E., Ladeveze, P., and Chinesta, F. (2017). Data-driven non-linear elasticity: constitutive manifold construction and problem discretization. *Computational Mechanics*, 60(5):813–826.

[Ibáñez et al., 2020] Ibáñez, R., Casteran, F., Argerich, C., Ghnatios, C., Hascoet, N., Ammar, A., Cassagnau, P., and Chinesta, F. (2020). On the data-driven modeling of reactive extrusion. *Fluids*, 5(2):94.

[Ibrahim, 2005] Ibrahim, R. A. (2005). *Liquid sloshing dynamics: theory and applications*. Cambridge University Press.

[Izidoro et al., 2007] Izidoro, D., Sierakowski, M.-R., Waszczynskyj, N., Haminiuk, C. W., and de Paula Scheer, A. (2007). Sensory evaluation and rheological behavior of commercial mayonnaise. *International journal of food engineering*, 3(1).

[Jain et al., 2021] Jain, P., Choudhury, A., Dutta, P., Kalita, K., Barsocchi, P., et al. (2021). Random forest regression-based machine learning model for accurate estimation of fluid flow in curved pipes. *Processes*, 9(11):2095.

[Jin et al., 2022] Jin, P., Zhang, Z., Kevrekidis, I. G., and Karniadakis, G. E. (2022). Learning poisson systems and trajectories of autonomous systems via poisson neural networks. *IEEE Transactions on Neural Networks and Learning Systems*.

[Jin et al., 2020] Jin, P., Zhu, A., Karniadakis, G. E., and Tang, Y. (2020). Symplectic networks: Intrinsic structure-preserving networks for identifying hamiltonian systems. *arXiv preprint arXiv:2001.03750*.

[Kaiser et al., 2018] Kaiser, E., Kutz, J. N., and Brunton, S. L. (2018). Discovering conservation laws from data for control. In *2018 IEEE Conference on Decision and Control (CDC)*, pages 6415–6421. IEEE.

[Kang et al., 2015] Kang, W., Zhang, J.-Z., Ren, S., and Lei, P.-F. (2015). Nonlinear galerkin method for low-dimensional modeling of fluid dynamic system using pod modes. *Communications in Nonlinear Science and Numerical Simulation*, 22(1-3):943–952.

[Kanno, 2018] Kanno, Y. (2018). Data-driven computing in elasticity via kernel regression. *Theoretical and Applied Mechanics Letters*, 8(6):361–365.

[Kant, 1908] Kant, I. (1908). Critique of pure reason. 1781. *Modern Classical Philosophers, Cambridge, MA: Houghton Mifflin*, pages 370–456.

[Kapteyn et al., 2020] Kapteyn, M. G., Willcox, K., and Knezevic, D. J. (2020). Toward predictive digital twins via component-based reduced-order models and interpretable machine learning. In *AIAA Scitech 2020 Forum*, page 0418.

[Karniadakis et al., 2021] Karniadakis, G. E., Kevrekidis, I. G., Lu, L., Perdikaris, P., Wang, S., and Yang, L. (2021). Physics-informed machine learning. *Nature Reviews Physics*, 3(6):422–440.

[Kashefi et al., 2021] Kashefi, A., Rempe, D., and Guibas, L. J. (2021). A point-cloud deep learning framework for prediction of fluid flow fields on irregular geometries. *Physics of Fluids*, 33(2):027104.

[Ke et al., 2019] Ke, N. R., Singh, A., Touati, A., Goyal, A., Bengio, Y., Parikh, D., and Batra, D. (2019). Learning dynamics model in reinforcement learning by incorporating the long term future. *arXiv preprint arXiv:1903.01599*.

[Keiper et al., 2018] Keiper, W., Milde, A., and Volkwein, S. (2018). *Reduced-order modeling (ROM) for simulation and optimization: powerful algorithms as key enablers for scientific computing*. Springer.

[Kennedy et al., 2019] Kennedy, M., Schmeckpeper, K., Thakur, D., Jiang, C., Kumar, V., and Daniilidis, K. (2019). Autonomous precision pouring from unknown containers. *IEEE Robotics and Automation Letters*, 4(3):2317–2324.

[Khaing and Masayuki, 2018] Khaing, M. P. and Masayuki, M. (2018). Transparent object detection using convolutional neural network. In *International Conference on Big Data Analysis and Deep Learning Applications*, pages 86–93. Springer.

[Kim et al., 2019] Kim, B., Azevedo, V. C., Thuerey, N., Kim, T., Gross, M., and Solenthaler, B. (2019). Deep fluids: A generative network for parameterized fluid simulations. In *Computer Graphics Forum*, volume 38, pages 59–70. Wiley Online Library.

[Kim et al., 2004] Kim, Y., Shin, Y.-S., and Lee, K. H. (2004). Numerical study on slosh-induced impact pressures on three-dimensional prismatic tanks. *Applied Ocean Research*, 26(5):213–226.

[Kingma and Ba, 2014] Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

[Kirchdoerfer and Ortiz, 2016] Kirchdoerfer, T. and Ortiz, M. (2016). Data-driven computational mechanics. *Computer Methods in Applied Mechanics and Engineering*, 304:81–101.

[Kirchdoerfer and Ortiz, 2018] Kirchdoerfer, T. and Ortiz, M. (2018). Data-driven computing in dynamics. *International Journal for Numerical Methods in Engineering*, 113(11):1697–1710.

[Koppal, 2014] Koppal, S. J. (2014). *Lambertian Reflectance*, pages 441–443. Springer US, Boston, MA.

[Kraus, 2021] Kraus, M. (2021). Metriplectic integrators for dissipative fluids. In *International Conference on Geometric Science of Information*, pages 292–301. Springer.

[Kroemer et al., 2021] Kroemer, O., Niekum, S., and Konidaris, G. D. (2021). A review of robot learning for manipulation: Challenges, representations, and algorithms. *Journal of machine learning research*, 22(30).

[Kubo, 1966] Kubo, R. (1966). The fluctuation-dissipation theorem. *Reports on progress in physics*, 29(1):255.

[Kumbár et al., 2018] Kumbár, V., Nedomová, Š., Ondrušíková, S., and Polcar, A. (2018). Rheological behaviour of chocolate at different temperatures. *Potravinarstvo*.

[Kutz et al., 2016] Kutz, J. N., Brunton, S. L., Brunton, B. W., and Proctor, J. L. (2016). *Dynamic mode decomposition: data-driven modeling of complex systems*. SIAM.

[Ladickỳ et al., 2015] Ladickỳ, L., Jeong, S., Solenthaler, B., Pollefeys, M., and Gross, M. (2015). Data-driven fluid simulations using regression forests. *ACM Transactions on Graphics (TOG)*, 34(6):1–9.

[Laroche and Barlier, 2017] Laroche, R. and Barlier, M. (2017). Transfer reinforcement learning with shared dynamics. In *Thirty-First AAAI Conference on Artificial Intelligence*.

[Lee et al., 2021] Lee, K., Trask, N., and Stinis, P. (2021). Machine learning structure preserving brackets for forecasting irreversible processes. *Advances in Neural Information Processing Systems*, 34.

[Lei et al., 2016] Lei, H., Baker, N. A., and Li, X. (2016). Data-driven parameterization of the generalized langevin equation. *Proceedings of the National Academy of Sciences*, 113(50):14183–14188.

[Li et al., 2018] Li, Y., Wu, J., Tedrake, R., Tenenbaum, J. B., and Torralba, A. (2018). Learning particle dynamics for manipulating rigid bodies, deformable objects, and fluids. *arXiv preprint arXiv:1810.01566*.

[Lieto et al., 2017] Lieto, A., Radicioni, D., Rho, V., and Mensa, E. (2017). Towards a unifying framework for conceptual represention and reasoning in cognitive systems. *Intelligenza Artificiale*, 11(2):139–153.

[Liu and Negrut, 2021] Liu, C. K. and Negrut, D. (2021). The role of physics-based simulators in robotics. *Annual Review of Control, Robotics, and Autonomous Systems*, 4:35–58.

[Liu and Wang, 2021] Liu, X.-Y. and Wang, J.-X. (2021). Physics-informed dyna-style model-based deep reinforcement learning for dynamic control. *Proceedings of the Royal Society A*, 477(2255):20210618.

[Liu and Tegmark, 2020] Liu, Z. and Tegmark, M. (2020). AI Poincaré: Machine Learning Conservation Laws from Trajectories. *arXiv preprint arXiv:2011.04698*.

[Lowe, 1999] Lowe, D. G. (1999). Object recognition from local scale-invariant features. In *Proceedings of the seventh IEEE international conference on computer vision*, volume 2, pages 1150–1157. Ieee.

[Lu et al., 2021a] Lu, L., Jin, P., Pang, G., Zhang, Z., and Karniadakis, G. E. (2021a). Learning nonlinear operators via deeponet based on the universal approximation theorem of operators. *Nature Machine Intelligence*, 3(3):218–229.

[Lu et al., 2021b] Lu, L., Pestourie, R., Yao, W., Wang, Z., Verdugo, F., and Johnson, S. G. (2021b). Physics-informed neural networks with hard constraints for inverse design. *SIAM Journal on Scientific Computing*, 43(6):B1105–B1132.

[Lui and Wolf, 2019] Lui, H. F. and Wolf, W. R. (2019). Construction of reduced-order models for fluid flows using deep feedforward neural networks. *Journal of Fluid Mechanics*, 872:963–994.

[Ly and Tran, 2001] Ly, H. V. and Tran, H. T. (2001). Modeling and control of physical processes using proper orthogonal decomposition. *Mathematical and computer modelling*, 33(1-3):223–236.

[Lye et al., 2020] Lye, K. O., Mishra, S., and Ray, D. (2020). Deep learning observables in computational fluid dynamics. *Journal of Computational Physics*, 410:109339.

[Ma et al., 2020] Ma, C., Wojtowytsch, S., Wu, L., et al. (2020). Towards a mathematical understanding of neural network-based machine learning: what we know and what we don't. *arXiv preprint arXiv:2009.10713*.

[Mao et al., 2020] Mao, Z., Jagtap, A. D., and Karniadakis, G. E. (2020). Physics-informed neural networks for high-speed flows. *Computer Methods in Applied Mechanics and Engineering*, 360:112789.

[Masi et al., 2020] Masi, F., Stefanou, I., Vannucci, P., and Maffi-Berthier, V. (2020). Material modeling via thermodynamics-based artificial neural networks. In *Workshop on Joint Structures and Common Foundations of Statistical Physics, Information Geometry and Inference for Learning*, pages 308–329. Springer.

[Masi et al., 2021] Masi, F., Stefanou, I., Vannucci, P., and Maffi-Berthier, V. (2021). Thermodynamics-based artificial neural networks for constitutive modeling. *Journal of the Mechanics and Physics of Solids*, 147:104277.

[Matl et al., 2019] Matl, C., Matthew, R., and Bajcsy, R. (2019). Haptic perception of liquids enclosed in containers. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 7142–7149. IEEE.

[Mielke, 2011] Mielke, A. (2011). Formulation of thermoelastic dissipative material behavior using generic. *Continuum Mechanics and Thermodynamics*, 23(3):233–256.

[Millán and Arroyo, 2013] Millán, D. and Arroyo, M. (2013). Nonlinear manifold learning for model reduction in finite elastodynamics. *Computer Methods in Applied Mechanics and Engineering*, 261:118–131.

[Miller et al., 2020] Miller, S. T., Lindner, J. F., Choudhary, A., Sinha, S., and Ditto, W. L. (2020). The scaling of physics-informed machine learning with data and dimensions. *Chaos, Solitons & Fractals: X*, 5:100046.

[Miyanawala and Jaiman, 2017] Miyanawala, T. P. and Jaiman, R. K. (2017). An efficient deep learning technique for the navier-stokes equations: Application to unsteady wake flow dynamics. *arXiv preprint arXiv:1710.09099*.

[Mohebujjaman et al., 2019] Mohebujjaman, M., Rebholz, L. G., and Iliescu, T. (2019). Physically constrained data-driven correction for reduced-order modeling of fluid flows. *International Journal for Numerical Methods in Fluids*, 89(3):103–122.

[Monaghan, 1992] Monaghan, J. J. (1992). Smoothed particle hydrodynamics. *Annual review of astronomy and astrophysics*, 30(1):543–574.

[Moreland et al., 2013] Moreland, J., Wang, J., Liu, Y., Li, F., Shen, L., Wu, B., and Zhou, C. (2013). Integration of augmented reality with computational fluid dynamics for power plant training. In *Proceedings of the International Conference on Modeling, Simulation and*

Visualization Methods (MSV), page 1. The Steering Committee of The World Congress in Computer Science, Computer ....

[Moriello et al., 2018] Moriello, L., Biagiotti, L., Melchiorri, C., and Paoli, A. (2018). Manipulating liquids with robots: A sloshing-free solution. *Control Engineering Practice*, 78:129–141.

[Morton et al., 2018] Morton, J., Jameson, A., Kochenderfer, M. J., and Witherden, F. (2018). Deep dynamical modeling and control of unsteady fluid flows. *Advances in Neural Information Processing Systems*, 31.

[Moya et al., 2020a] Moya, B., Alfaro, I., Gonzalez, D., Chinesta, F., and Cueto, E. (2020a). Physically sound, self-learning digital twins for sloshing fluids. *PLoS One*, 15(6):e0234569.

[Moya et al., 2020b] Moya, B., Badías, A., Alfaro, I., Chinesta, F., and Cueto, E. (2020b). Digital twins that learn and correct themselves. *International Journal for Numerical Methods in Engineering*.

[Moya et al., 2019] Moya, B., Gonzalez, D., Alfaro, I., Chinesta, F., and Cueto, E. (2019). Learning slosh dynamics by means of data. *Computational Mechanics*, 64(2):511–523.

[Mrowca et al., 2018] Mrowca, D., Zhuang, C., Wang, E., Haber, N., Fei-Fei, L. F., Tenenbaum, J., and Yamins, D. L. (2018). Flexible neural representation for physics prediction. *Advances in neural information processing systems*, 31.

[Müller et al., 2003] Müller, M., Charypar, D., and Gross, M. (2003). Particle-based fluid simulation for interactive applications. In *Proceedings of the 2003 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 154–159. Citeseer.

[Munch, 2017] Munch, E. (2017). A user's guide to topological data analysis. *Journal of Learning Analytics*, 4(2):47–61.

[Mur-Artal et al., 2015] Mur-Artal, R., Montiel, J. M. M., and Tardos, J. D. (2015). Orb-slam: a versatile and accurate monocular slam system. *IEEE transactions on robotics*, 31(5):1147–1163.

[Murata et al., 2020] Murata, T., Fukami, K., and Fukagata, K. (2020). Nonlinear mode decomposition with convolutional neural networks for fluid dynamics. *Journal of Fluid Mechanics*, 882.

[Nair et al., 2017] Nair, A., Chen, D., Agrawal, P., Isola, P., Abbeel, P., Malik, J., and Levine, S. (2017). Combining self-supervised learning and imitation for vision-based rope manipulation. In *2017 IEEE international conference on robotics and automation (ICRA)*, pages 2146–2153. IEEE.

[Nava et al., 2021] Nava, M., Paolillo, A., Guzzi, J., Gambardella, L. M., and Giusti, A. (2021). Uncertainty-aware self-supervised learning of spatial perception tasks. *arXiv preprint arXiv:2103.12007*.

[Ng et al., 2011] Ng, A. et al. (2011). Sparse autoencoder. *CS294A Lecture notes*, 72(2011):1–19.

[Nguyen and Turski, 2001] Nguyen, S. Q. and Turski, Ł. A. (2001). On the dirac approach to constrained dissipative dynamics. *Journal of Physics A: Mathematical and General*, 34(43):9281.

[Novati et al., 2021] Novati, G., de Laroussilhe, H. L., and Koumoutsakos, P. (2021). Automating turbulence modelling by multi-agent reinforcement learning. *Nature Machine Intelligence*, 3(1):87–96.

[Oh et al., 2015] Oh, J., Guo, X., Lee, H., Lewis, R., and Singh, S. (2015). Action-conditional video prediction using deep networks in atari games. *arXiv preprint arXiv:1507.08750*.

[Oishi and Yagawa, 2017] Oishi, A. and Yagawa, G. (2017). Computational mechanics enhanced by deep learning. *Computer Methods in Applied Mechanics and Engineering*, 327:327–351.

[Ortali et al., 2020] Ortali, G., Demo, N., and Rozza, G. (2020). Gaussian process approach within a data-driven pod framework for fluid dynamics engineering problems. *arXiv preprint arXiv:2012.01989*.

[Öttinger, 2015] Öttinger, H. C. (2015). Preservation of thermodynamic structure in model reduction. *Physical Review E*, 91(3):032147.

[Pan et al., 2016] Pan, Z., Park, C., and Manocha, D. (2016). Robot motion planning for pouring liquids. In *Twenty-Sixth International Conference on Automated Planning and Scheduling*.

[Pang et al., 2019] Pang, G., Yang, L., and Karniadakis, G. E. (2019). Neural-net-induced gaussian process regression for function approximation and pde solution. *Journal of Computational Physics*, 384:270–288.

[Parr et al., 2008] Parr, R., Li, L., Taylor, G., Painter-Wakefield, C., and Littman, M. L. (2008). An analysis of linear models, linear value-function approximation, and feature selection for reinforcement learning. In *Proceedings of the 25th international conference on Machine learning*, pages 752–759.

[Pascanu et al., 2013] Pascanu, R., Mikolov, T., and Bengio, Y. (2013). On the difficulty of training recurrent neural networks. In *International conference on machine learning*, pages 1310–1318. PMLR.

[Portillo et al., 2017] Portillo, D., García Orden, J., and Romero, I. (2017). Energy–entropy–momentum integration schemes for general discrete non-smooth dissipative problems in thermomechanics. *International Journal for Numerical Methods in Engineering*, 112(7):776–802.

[Proctor et al., 2016] Proctor, J. L., Brunton, S. L., and Kutz, J. N. (2016). Dynamic mode decomposition with control. *SIAM Journal on Applied Dynamical Systems*, 15(1):142–161.

[Proctor et al., 2018] Proctor, J. L., Brunton, S. L., and Kutz, J. N. (2018). Generalizing koopman theory to allow for inputs and control. *SIAM Journal on Applied Dynamical Systems*, 17(1):909–930.

[Rabault and Kuhnle, 2019] Rabault, J. and Kuhnle, A. (2019). Accelerating deep reinforcement learning strategies of flow control through a multi-environment approach. *Physics of Fluids*, 31(9):094105.

[Raissi and Karniadakis, 2018] Raissi, M. and Karniadakis, G. E. (2018). Hidden physics models: Machine learning of nonlinear partial differential equations. *Journal of Computational Physics*, 357:125–141.

[Raissi et al., 2017] Raissi, M., Perdikaris, P., and Karniadakis, G. E. (2017). Physics informed deep learning (part i): Data-driven solutions of nonlinear partial differential equations. *arXiv preprint arXiv:1711.10561*.

[Raissi et al., 2018] Raissi, M., Perdikaris, P., and Karniadakis, G. E. (2018). Numerical gaussian processes for time-dependent and nonlinear partial differential equations. *SIAM Journal on Scientific Computing*, 40(1):A172–A198.

[Ranjan et al., 2018] Ranjan, A., Bolkart, T., Sanyal, S., and Black, M. J. (2018). Generating 3d faces using convolutional mesh autoencoders. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 704–720.

[Rao et al., 2021] Rao, C., Sun, H., and Liu, Y. (2021). Physics-informed deep learning for computational elastodynamics without labeled data. *Journal of Engineering Mechanics*, 147(8):04021043.

[Rath et al., 2022] Rath, L., Geist, A. R., and Trimpe, S. (2022). Using physics knowledge for learning rigid-body forward dynamics with gaussian process force priors. In *Conference on Robot Learning*, pages 101–111. PMLR.

[Recasens et al., 2021] Recasens, D., Lamarca, J., Fácil, J. M., Montiel, J., and Civera, J. (2021). Endo-depth-and-motion: Reconstruction and tracking in endoscopic videos using depth networks and photometric constraints. *IEEE Robotics and Automation Letters*, 6(4):7225–7232.

[Ren et al., 2021] Ren, F., Rabault, J., and Tang, H. (2021). Applying deep reinforcement learning to active flow control in weakly turbulent conditions. *Physics of Fluids*, 33(3):037121.

[Reynolds et al., 2007] Reynolds, C. R., Fletcher-Janzen, E., et al. (2007). *Encyclopedia of Special Education: A Reference for the Education of Children, Adolescents, and Adults with Disabilities and Other Exceptional Individuals, Volume 3*, volume 3. John Wiley & Sons.

[Rezende et al., 2014] Rezende, D. J., Mohamed, S., and Wierstra, D. (2014). Stochastic backpropagation and approximate inference in deep generative models. In *International conference on machine learning*, pages 1278–1286. PMLR.

[Rodríguez-Ocampo et al., 2020] Rodríguez-Ocampo, P., Ring, M., Hemandez-Fontes, J., Alcérreca-Huerta, J., Mendoza, E., Gallegos-Diez-Barroso, G., and Silva, R. (2020). A 2d image-based approach for cfd validation of liquid mixing in a free-surface condition. *Journal of Applied Fluid Mechanics*, 13(5).

[Romero, 2009a] Romero, I. (2009a). Thermodynamically consistent time-stepping algorithms for non-linear thermomechanical systems. *International journal for numerical methods in engineering*, 79(6):706–732.

[Romero, 2009b] Romero, I. (2009b). Thermodynamically consistent time-stepping algorithms for non-linear thermomechanical systems. *International Journal for Numerical Methods in Engineering*, 79(6):706–732.

[Romero, 2010] 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. *Computer Methods in Applied Mechanics and Engineering*, 199(25-28):1841–1858.

[Romero, 2013] Romero, I. (2013). A characterization of conserved quantities in non-equilibrium thermodynamics. *Entropy*, 15(12):5580–5596.

[Roweis and Saul, 2000] Roweis, S. T. and Saul, L. K. (2000). Nonlinear dimensionality reduction by locally linear embedding. *science*, 290(5500):2323–2326.

[Rowley, 2005] Rowley, C. W. (2005). Model reduction for fluids, using balanced proper orthogonal decomposition. *International Journal of Bifurcation and Chaos*, 15(03):997–1013.

[Rowley and Dawson, 2017] Rowley, C. W. and Dawson, S. T. (2017). Model reduction for flow analysis and control. *Annual Review of Fluid Mechanics*, 49:387–417.

[Sajjan et al., 2020] Sajjan, S., Moore, M., Pan, M., Nagaraja, G., Lee, J., Zeng, A., and Song, S. (2020). Clear grasp: 3d shape estimation of transparent objects for manipulation. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3634–3642. IEEE.

[Sajjan et al., 2019] Sajjan, S. S., Moore, M., Pan, M., Nagaraja, G., Lee, J., Zeng, A., and Song, S. (2019). Cleargrasp: 3d shape estimation of transparent objects for manipulation. *arXiv preprint arXiv:1910.02550*.

[Sancarlos et al., 2021a] Sancarlos, A., Cameron, M., Abel, A., Cueto, E., Duval, J.-L., and Chinesta, F. (2021a). From rom of electrochemistry to ai-based battery digital and hybrid twin. *Archives of Computational Methods in Engineering*, 28(3):979–1015.

[Sancarlos et al., 2021b] Sancarlos, A., Champaney, V., Duval, J.-L., Cueto, E., and Chinesta, F. (2021b). Pgd-based advanced nonlinear multiparametric regressions for constructing metamodels at the scarce-data limit. *arXiv preprint arXiv:2103.05358*.

[Sancarlos et al., 2021c] Sancarlos, A., Cueto, E., Chinesta, F., and Duval, J.-L. (2021c). A novel sparse reduced order formulation for modeling electromagnetic forces in electric motors. *SN Applied Sciences*, 3(3):1–19.

[Sanchez-Gonzalez et al., 2020] Sanchez-Gonzalez, A., Godwin, J., Pfaff, T., Ying, R., Leskovec, J., and Battaglia, P. (2020). Learning to simulate complex physics with graph networks. In *International Conference on Machine Learning*, pages 8459–8468. PMLR.

[Schenck and Fox, 2016a] Schenck, C. and Fox, D. (2016a). Detection and tracking of liquids with fully convolutional networks. *arXiv preprint arXiv:1606.06266*.

[Schenck and Fox, 2016b] Schenck, C. and Fox, D. (2016b). Towards learning to perceive and reason about liquids. In *International Symposium on Experimental Robotics*, pages 488–501. Springer.

[Schenck and Fox, 2018a] Schenck, C. and Fox, D. (2018a). Perceiving and reasoning about liquids using fully convolutional networks. *The International Journal of Robotics Research*, 37(4-5):452–471.

[Schenck and Fox, 2018b] Schenck, C. and Fox, D. (2018b). Spnets: Differentiable fluid dynamics for deep neural networks. In *Conference on Robot Learning*, pages 317–335. PMLR.

[Schmidt and Lipson, 2009] Schmidt, M. and Lipson, H. (2009). Distilling free-form natural laws from experimental data. *science*, 324(5923):81–85.

[Schölkopf et al., 1998] Schölkopf, B., Smola, A., and Müller, K.-R. (1998). Nonlinear component analysis as a kernel eigenvalue problem. *Neural computation*, 10(5):1299–1319.

[Shi et al., 1994] Shi, J. et al. (1994). Good features to track. In *1994 Proceedings of IEEE conference on computer vision and pattern recognition*, pages 593–600. IEEE.

[Shvartsman and Kevrekidis, 1998] Shvartsman, S. Y. and Kevrekidis, I. G. (1998). Nonlinear model reduction for control of distributed systems: A computer-assisted study. *AIChE Journal*, 44(7):1579–1595.

[Stainier et al., 2019] Stainier, L., Leygue, A., and Ortiz, M. (2019). Model-free data-driven methods in mechanics: material data identification and solvers. *Computational Mechanics*, 64(2):381–393.

[Su and Yang, 2002] Su, S.-F. and Yang, F.-Y. (2002). On the dynamical modeling with neural fuzzy networks. *IEEE Transactions on Neural Networks*, 13(6):1548–1553.

[Sun and Wang, 2020] Sun, L. and Wang, J.-X. (2020). Physics-constrained bayesian neural network for fluid flow reconstruction with sparse and noisy data. *Theoretical and Applied Mechanics Letters*, 10(3):161–169.

[Sutton and Barto, 2018] Sutton, R. S. and Barto, A. G. (2018). *Reinforcement learning: An introduction*. MIT press.

[Taira et al., 2020] Taira, K., Hemati, M. S., Brunton, S. L., Sun, Y., Duraisamy, K., Bagheri, S., Dawson, S. T., and Yeh, C.-A. (2020). Modal analysis of fluid flows: Applications and outlook. *AIAA journal*, 58(3):998–1022.

[Taylor and Stone, 2009] Taylor, M. E. and Stone, P. (2009). Transfer learning for reinforcement learning domains: A survey. *Journal of Machine Learning Research*, 10(7).

[Tenenbaum et al., 2000] Tenenbaum, J. B., Silva, V. d., and Langford, J. C. (2000). A global geometric framework for nonlinear dimensionality reduction. *science*, 290(5500):2319–2323.

[Tezzele et al., 2018] Tezzele, M., Demo, N., Mola, A., and Rozza, G. (2018). An integrated data-driven computational pipeline with model order reduction for industrial and applied mathematics. *arXiv preprint arXiv:1810.12364*.

[Tompson et al., 2017] Tompson, J., Schlachter, K., Sprechmann, P., and Perlin, K. (2017). Accelerating eulerian fluid simulation with convolutional networks. In *International Conference on Machine Learning*, pages 3424–3433. PMLR.

[Tosun et al., 2017] Tosun, U., Aghazadeh, R., Sert, C., and Özer, M. B. (2017). Tracking free surface and estimating sloshing force using image processing. *Experimental Thermal and Fluid Science*, 88:423–433.

[Toth et al., 2019] Toth, P., Rezende, D. J., Jaegle, A., Racanière, S., Botev, A., and Higgins, I. (2019). Hamiltonian generative networks. *arXiv preprint arXiv:1909.13789*.

[Treuille et al., 2006] Treuille, A., Lewis, A., and Popović, Z. (2006). Model reduction for real-time fluids. *ACM Transactions on Graphics (TOG)*, 25(3):826–834.

[Ummenhofer et al., 2019] Ummenhofer, B., Prantl, L., Thuerey, N., and Koltun, V. (2019). Lagrangian fluid simulation with continuous convolutions. In *International Conference on Learning Representations*.

[Vázquez-Quesada et al., 2009] Vázquez-Quesada, A., Ellero, M., and Español, P. (2009). Consistent scaling of thermal fluctuations in smoothed dissipative particle dynamics. *The Journal of chemical physics*, 130(3):034901.

[Verma et al., 2018] Verma, S., Novati, G., and Koumoutsakos, P. (2018). Efficient collective swimming by harnessing vortices through deep reinforcement learning. *Proceedings of the National Academy of Sciences*, 115(23):5849–5854.

[Vlassis and Sun, 2021] Vlassis, N. N. and Sun, W. (2021). Sobolev training of thermodynamic-informed neural networks for interpretable elasto-plasticity models with level set hardening. *Computer Methods in Applied Mechanics and Engineering*, 377:113695.

[Wang et al., 2005] Wang, J., Hertzmann, A., and Fleet, D. J. (2005). Gaussian process dynamical models. *Advances in neural information processing systems*, 18.

[Wang et al., 2007] Wang, J. M., Fleet, D. J., and Hertzmann, A. (2007). Gaussian process dynamical models for human motion. *IEEE transactions on pattern analysis and machine intelligence*, 30(2):283–298.

[Wang et al., 2019a] Wang, T., Bao, X., Clavera, I., Hoang, J., Wen, Y., Langlois, E., Zhang, S., Zhang, G., Abbeel, P., and Ba, J. (2019a). Benchmarking model-based reinforcement learning. *arXiv preprint arXiv:1907.02057*.

[Wang and Carreira-Perpinan, 2014] Wang, W. and Carreira-Perpinan, M. (2014). The role of dimensionality reduction in classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 28.

[Wang et al., 2019b] Wang, Y., Shen, Z., Long, Z., and Dong, B. (2019b). Learning to discretize: solving 1d scalar conservation laws via deep reinforcement learning. *arXiv preprint arXiv:1905.11079*.

[Wasserman, 2018] Wasserman, L. (2018). Topological data analysis. *Annual Review of Statistics and Its Application*, 5:501–532.

[Weinan, 2017] Weinan, E. (2017). A proposal on machine learning via dynamical systems. *Communications in Mathematics and Statistics*, 1(5):1–11.

[Welling and Teh, 2011] Welling, M. and Teh, Y. W. (2011). Bayesian learning via stochastic gradient langevin dynamics. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, pages 681–688. Citeseer.

[Wiewel et al., 2019] Wiewel, S., Becher, M., and Thuerey, N. (2019). Latent space physics: Towards learning the temporal evolution of fluid flow. In *Computer graphics forum*, volume 38, pages 71–82. Wiley Online Library.

[Williams, 1998] Williams, C. K. (1998). Prediction with gaussian processes: From linear regression to linear prediction and beyond. In *Learning in graphical models*, pages 599–621. Springer.

[Wu et al., 2015] Wu, J., Yildirim, I., Lim, J. J., Freeman, B., and Tenenbaum, J. (2015). Galileo: Perceiving physical object properties by integrating a physics engine with deep learning. *Advances in neural information processing systems*, 28:127–135.

[Wu et al., 2020] Wu, K., Qin, T., and Xiu, D. (2020). Structure-preserving method for reconstructing unknown hamiltonian systems from trajectory data. *SIAM Journal on Scientific Computing*, 42(6):A3704–A3729.

[Xiao et al., 2014] Xiao, D., Fang, F., Buchan, A. G., Pain, C. C., Navon, I. M., Du, J., and Hu, G. (2014). Non-linear model reduction for the navier–stokes equations using residual deim method. *Journal of Computational Physics*, 263:1–18.

[Xie et al., 2020] Xie, X., Webster, C., and Iliescu, T. (2020). Closure learning for nonlinear model reduction using deep residual neural network. *Fluids*, 5(1):39.

[Yan et al., 2020] Yan, M., Zhu, Y., Jin, N., and Bohg, J. (2020). Self-supervised learning of state estimation for manipulating deformable linear objects. *IEEE robotics and automation letters*, 5(2):2372–2379.

[Yang et al., 2021] Yang, S., Wong, S. W., and Kou, S. (2021). Inference of dynamic systems from noisy and sparse data via manifold-constrained gaussian processes. *Proceedings of the National Academy of Sciences*, 118(15).

[Yang et al., 2018] Yang, X., Tartakovsky, G., and Tartakovsky, A. (2018). Physics-informed kriging: A physics-informed gaussian process regression method for data-model convergence. *arXiv preprint arXiv:1809.03461*.

[Yang and Perdikaris, 2019] Yang, Y. and Perdikaris, P. (2019). Conditional deep surrogate models for stochastic, high-dimensional, and multi-fidelity systems. *Computational Mechanics*, 64(2):417–434.

[Yildirim et al., 2015] Yildirim, I., Wu, J., Lim, J. J., Tenenbaum, J. B., and Freeman, W. T. (2015). Galileo: Perceiving physical object properties by integrating a physics engine with deep learning.

[Yu et al., 2021] Yu, H., Tian, X., Weinan, E., and Li, Q. (2021). Onsagernet: Learning stable and interpretable dynamics using a generalized onsager principle. *Physical Review Fluids*, 6(11):114402.

[Zames et al., 1981] Zames, G., Ajlouni, N., Ajlouni, N., Ajlouni, N., Holland, J., Hills, W., and Goldberg, D. (1981). Genetic algorithms in search, optimization and machine learning. *Information Technology Journal*, 3(1):301–302.

[Zhang et al., 2018] Zhang, C., Bütepage, J., Kjellström, H., and Mandt, S. (2018). Advances in variational inference. *IEEE transactions on pattern analysis and machine intelligence*, 41(8):2008–2026.

[Zhang et al., 2021] Zhang, Z., Shin, Y., and Karniadakis, G. E. (2021). Gfinns: Generic formalism informed neural networks for deterministic and stochastic dynamical systems. *arXiv preprint arXiv:2109.00092*.

[Zhong et al., 2019] Zhong, Y. D., Dey, B., and Chakraborty, A. (2019). Symplectic ode-net: Learning hamiltonian dynamics with control. *arXiv preprint arXiv:1909.12077*.