TESIS DE LA UNIVERSIDAD DE ZARAGOZA

2022 108

Carmen Eugenia Alfaro Isac

Strategies of data analysis for numerical simulation of industrial processes

Director/es
Izquierdo Estallo, Salvador

ISSN 2254-7606

Prensas de la Universidad
Universidad Zaragoza

ISSN 2254-7606

Universidad Zaragoza
1542

Tesis Doctoral

STRATEGIES OF DATA ANALYSIS FOR NUMERICAL SIMULATION OF INDUSTRIAL PROCESSES

Autor

Carmen Eugenia Alfaro Isac

Director/es

Izquierdo Estallo, Salvador

UNIVERSIDAD DE ZARAGOZA
Escuela de Doctorado

Programa de Doctorado en Mecánica de Fluidos

2022

UNIVERSIDAD DE ZARAGOZA

DOCTORAL THESIS

Strategies of data analysis for numerical simulation of industrial processes

---

Author:
Carmen ALFARO ISAC

Supervisor:
Dr. Salvador IZQUIERDO ESTALLO

"If we assume we've arrived: we stop searching, we stop developing. "

Jocelyn Bell Burnell, astrophysicist.

UNIVERSIDAD DE ZARAGOZA

Abstract

Escuela de Ingeniería y Arquitectura de Zaragoza (EINA)

Departmento de Ciencia y Tecnología de Materiales y Fluidos

Doctor of Philosophy

Strategies of data analysis for numerical simulation of industrial processes

by Carmen ALFARO ISAC

Industrial digitalization and the seek for competitiveness has increased the demand for efficient online application for optimization of control parameters, quality prediction and defect prevention. This can be achieved through Digital Twins, which provide the digital-physical integration of manufacturing processes. To be suitable for industrial implementation, the virtual model must be fast, robust and accurate.

Numerical simulations are very useful to gain a deeper understanding of physical systems and create their corresponding interpretable models. However, the high computational cost hinders their direct implementation in industrial environments. Hence, this work aims to provide coupled strategies of numerical simulation and data analysis that allow their integration in online predictive applications.

Both tools can be combined using different levels of process-data utilization to develop several process models. These approaches are tested using a rubber compounding case. A theoretical model is presented that provides an exhaustive comprehension of the process. Data-driven models are more accurate but only feature-selected dimensionality reduction techniques allow to preserve the physical interpretability. Two additional industrial problems are assessed using hybrid models. Thus, the modelling of manufacturing processes be tackled using different approaches and the combination of data-based and physical models allows for the creation of accurate, fast and interpretable models for industrial deployment.

In addition, the performance of models and computer simulation can be improved by upgrading the coupling strategies among them. In this work, a ROM implementation in a commercial CFD code leads to a robust and efficient calculation of thermodynamic properties. The complex computation of equations of state is replaced by an equivalent accurate ROM.

Moreover, the division of a manifold into smaller subdomains of particular characteristics that are individually fit to a ROM increases the accuracy of the global prediction. This can be done using knowledge-guided separation or data-driven automatic division.

The selection of the design of experiments to generate the training dataset to build a ROM with is also critical for the performance of the ROM. A sequential sampling algorithm is proposed for the efficient calculation of sampling datasets. To achieve that, the available information of the response of the system is included to compute the data point that should be sampled next.

Acknowledgements

First of all, I would like to thank my project advisor Salvador Izquierdo for offering me to start a PhD under his supervision. When I accepted I was unaware of the enormous impact it would have on my professional and personal development in the next years. Moreover, his technical guidance has been key to the success of this research.

I would also like to thank my colleagues, who have provided me with their useful expertise any time I requested it. I equally value a deep scientific knowledge as much as an open attitude for listening and patience, and I feel extremely fortunate to have found people during this time that has both qualities.

To my family, especially my parents, whose unconditional support can not be put into words. To my friends, for encouraging me in moments of distress and joining me in celebration of even the smallest achievements.

Contents

List of Figures

List of Tables

Chapter 1

Background and motivation

1.1 The role of numerical simulation in the Digital Twin era

1.1.1 A historical perspective of numerical simulation

The beginning of modern numerical analysis dates back to World War II, when computational simulations were required to the successful design of the nuclear weapons at the Manhattan Project. Their design depended on solving the complex calculations regarding the motion of neutrons. For that purpose, the two mathematicians Jon Von Neumann and Stanislaw Ulam invented the Montecarlo method and use it to simulate the behaviour of neutrons on ENIAC [17], one of the first digital computers [81, 84, 103].

Hence, though the first computer simulation was developed as a mathematical tool in the nuclear physics field, it quickly expanded toengineering, physics, chemistry, biology and climate sciences, among other disciplines [15].

Even though numerical simulation is the only tool to study complex physical problems, it did not become relevant until the '90s. The delay of its use was mainly caused by the lack of both skilled people and computer resources. It is important to remember that at the early stages of digital computers development, the available programming languages were very limited and the time and effort to develop and run a single computer simulation were enormous [105]. In addition, the bibliography about the topic was limited, so the fundamentals of computer programming was mainly based on their own experience from practice or instructor's experience, which prevented the increase of experts in the field [88].

The advances in computer technology were essential for the popularization of numerical simulation as a problem-solving method. The increase of processor speed and memory sizes along with the decrease of the size of the components and the reduction of the costs facilitated the general access to computers. Hardware progresses allowed the development of simulation software systems that could perform numerical studies with affordable computational resources and time [88].

Consequently, the revolution in digital computing led to the rapid growth of numerical simulation techniques. Significant developments in simulation methodologies such as the output analysis [87], the design of experiments [38], optimization [35] or verification and validation [62] have been reported since.

The general aim of computer simulations is to understand physical systems and reproduce and study their behaviour. This concept includes the exploration of the physical system itself and the interaction among the different elements, as well as the calculation of the expected response of the system under different physical scenarios [131]. Thus it constitutes a key tool of scientific research. Unfortunately, its applicability in manufacturing is restricted to the industrial design and the offline optimization of products and processes.

When it is necessary to explore the dynamic of fluid systems, computer simulation becomes especially relevant. The complexity of the governing equations entails that the discretization of the corresponding differential equations and its posterior resolution using step-by-step methods of numerical simulation are the only available tool to explore these fluid systems.

1.1.2 The emerging Digital Twin

The rapid development of computer technologies described in the previous section also drove the digitalization of the manufacturing industry. Hence, the advances in process automation and integration, data acquisition and data communication promoted a closer connection between the physical world and the virtual space. Nowadays, this physic-virtual connection is often referred by the Digital Twin term.

Many references agree to indicate that the concept of Digital Twin as a virtual representation of physical entities was first introduced by Michael Grieves in 2003 under the denomination of “Conceptual Ideal for Product Lifecycle

Management" [44]. After a few different names, the first mention of Digital Twin originated in the Roadmap Report of NASA in 2010 as "an integrated multiphysics, multiscale, probabilistic simulation of an as-built vehicle or system that uses the best available physical models, sensor updates, fleet history, etc" [42].

The term quickly expanded from the aerospace industry to other research areas and production systems. Digital twins provide the efficient synchronization between the physical world and their corresponding virtual models demanded by the Smart Industry. The digital-physical integration of material models and transformation processes allows the virtual prediction of the evolution of the product along the manufacturing line. Thus, it becomes a powerful method to ensure product quality and to prevent the appearance and defects propagation. The combination of Digital Twins and the multi-source data gathering, Internet of Things (IoT) systems, and Virtual and Augmented reality (VR/AR) environments will result in highly effective manufacturing planning and precise production control [98].

In the earlier years, although most papers proposed definitions of the Digital Twin closer to a high-fidelity model or multidisciplinary simulation, they ommitted the real-time connection of the virtual model to the physical object. As research on the topic evolved, the dynamic and bidirectional mapping to the real process became more relevant [74]. This interactionenables the control of simulation-based engineering and applications in real-time. Digital Twin shows, therefore, huge potential for enhancing manufacturing systems.

As the interest in Digital Twin grew, so did the diversity of understandings of the concept. This led to a wide variety of definitions of Digital Twin. Multiple visions can be found in literature reviews across industries [66, 118, 89, 132]. The lack of a comprehensive and in-depth analysis of Digital Twin from the perspective of concepts, technologies and industrial applications complicates the establishment of a unified description in terms of boundary determination, implementation framework or protocols. Unfortunately, the disagreements among experts across different scientific disciplines or industrial sectors are an obstacle to the acceptance a common Digital Twin concept in manufacturing. To address this, currently some standards for Digital Twin are being developed, such as ISO23247: Digital twin framework for manufacturing [106].

In this dissertation, the author has considered that the Digital Twin is defined according to three different stages which are sorted according to the

level of the process virtualisation. Moreover, they usually coincide with the sophistication of the digital twins: physical, virtualised and twin thread.

The physical stage covers the real-time or near real-time applications of the IoT in the manufacturing processes of the industry through the measurements of sensors, gauges, RFID tags and readers, cameras, scanners, etc. At this stage, the data that Digital Twin needs are usually of vast volume, high velocity, and large variety, being difficult and costly to transmit to Digital Twin in the cloud server. Thus, to reduce the network burden, Artifical Intelligence (AI) and machine learning (ML) algorithms are ideal methods to pre-process the collected data.

The virtualised stage of Digital Twin uses in near real-time the results of the physics-based simulation models to augment the information about the manufacturing processes. Contrary to the traditional computer simulation of the process, the Digital Twin in this stage uses real-time data from the physical system that is collected and recorded from the physical space via IoT sensors [117].

After adequate treatment and processing through machine learning techniques, data is converted into a data-driven corrective model. This the model can be then incorporated into the Digital Twin to continuously improve its accuracy, progressively filling the gap between the digital and the physical worlds and transforming it into a smart Digital Twin. In this twin-thread stage, the Digital Twin can recognize, analyze and support decision-making through the implementation of AI and ML algorithms. Digital Twin in this level represents an in-depth integration of new-generation artificial intelligence technology and advanced manufacturing to obtain powerful learning and cognitive capacities.

1.1.3 Digital Twin and numerical simulation

For an efficient representation of the real system, it is strongly recommended that the Digital Twin is quick, accurate and physics-based [132].

As stated in the previous section, computer-based simulation models are widely used in the stages of product and process design and offline optimization. Numerical simulation allows designers to understand in depth the physical behaviour of a product or a process, detect and correct errors and malfunctions, predict and analyze virtual what-if scenarios, fine-tune the design of the system before it is built and thereby drastically reduce the number

of physical prototypes and physical tests needed to design and prepare a new product for production. Computer-aided design, dynamic simulation, finite element models, computational fluid dynamics models, and even multi-domain models with fluid-structure, thermal-structure, and other types of multiphasic interactions are routinely used to this effect. Multi-physics and multi-scale simulations are the digital tools that provide the most accurate and complete model of a physical system.

It is clear that the information generated by computationally expensive physics-based simulations is highly valuable but the deployment of high-fidelity simulations is unfeasible in an industrial environment where real-time responses are required for decision-making. Nevertheless, that does not mean that computer-based simulation models should be completely discarded in the manufacturing industry.

The challenge is to integrate numerical simulation in the Digital Twin reducing the computational time to provide a fast and precise physics-based response to a specific process in near-real-time. To instantaneously obtain the ouptut of the system from a given set of input parameters, it is necessary to have the solution for any value of the spatial coordinates, time, and material, geometrical, and load parameters.

The pre-computing of all the possible solutions to a problem is however unfeasible, as the number of scenarios increases exponentially with the number of parameters and the range of parameter values. To reduce the processing data Reduction Order Models (ROMs) are able to transform complex and computationally costly Digital Twins into real-time applications that can be integrated into production environments.

1.2 Model classification

The model update of a Digital Twin is always data-driven, since the inputs are measured quantities from the physical system, like material properties, boundary conditions or loads. The Digital Twin predictions must also be directly equivalent to a measured quantity [96].

However, that does not mean that necessarily the Digital Twin model is data-based. Thanks to the increasing sensorization of production systems and the embracing of the Industrial Internet of Things (IIoT), data amount and availability from manufacturing processes grew. Consequently, data-based

models, also called black-box models, became more popular. They are solely inferred from the relation among the system's inputs and output, which eliminates the need for introducing knowledge of the process and its associated costs (experts teams, advanced computer software). They are widely used in manufacturing processes for optimization of control parameters [76, 68, 48, 43, 5], quality prediction of a processed material or manufactured product [77, 75], among others.

The generation of data-driven models comprises three steps:

- Data collection: selecting the appropriate sensors in the process is critical for a successful data-driven model. If they are not already installed, the mechanical integration of these sensors is also part of this stage, as well as the selection of the communication technology from the sensors to the data storage.
- Data processing: it refers to the development of a solid and reliable infrastructure for data storage, communication and computation. This step includes the determination of the location of data processing and information providing, along with the selection of network protocols, IIoT gateways and the definition of data security.
- Data analysis: it is usually necessary to perform some data transformation techniques over the collected data, such as data cleaning, filtering or feature extraction, to prepare it or improve it before creating the data-driven model using ML techniques.

The efficiency of the data-based models strongly depends on the quality of the input data, which can be limited by the number or location of sensors, the distorsion caused by signal noise or even the corruption due to process malfunctions. In addition, they miss the causal relationships among the involved process variables; thus they lack physical interpretability. Consequently, these models are not extrapolative and they are only reliable within the range of data.

On the other hand, physics-based models or white-box models are able to simulate complex physical systems replicating the predominant phenomena, allowing for a deeper understanding of the process. These models can provide the Digital Twin with physical interpretability, which means that it could generate a robust prediction even in unlikely scenarios.

The development of a physics-based model is divided into three phases:

- Understanding of the physical system: the real process or product is thoroughly reviewed in order to identify the relevant mechanisms and the critical scenarios.
- Model definition: the preliminary formulation of the model is established, as well as the system parameters and boundaries. Also at this stage, the required information, if it is available, is collected, such as material specifications and datasheets of the product or different components of the production systems.
- Model creation: the simulation method and simulation software is firstly selected. A design of experiments is conducted and the corresponding computer simulations are executed. Finally, a careful result analysis is performed, which includes results visualization and interpretation.

However, the requirements of the white-box models are exigent. They usually demand specific software, a team of experts and expensive computational resources, which hinders their implementation in industrial applications.

A summary of the characteristics of each model type is shown in Table 1.1.

TABLE 1.1: Main features of white-box and black-box models.

White-box	**Black-box**
Knowledge-based	Data-based
Theoretical analysis	Statistical analysis
Computationally expensive	Quick
Extrapolative and interpretable	Restricted to domain range
Idealized by assumptions	Data inaccuracies migh be unnoticed

A third type of model has emerged that combines data-driven and physics-based models. They are denoted as hybrid or grey-box models and aim at overcoming the limitations of their predecessors. Ideally, these models include general or partial physical information and thus are more extrapolative and adaptative than data-based models while demanding fewer amounts of data or process variables. They are easier to develop compared to physics-based models and allow the construction of models even in the presence of very complex physical phenomena [138, 125]. Moreover, due to their flexibility and capabilities, hybrid models seem to be a promising approach to create efficient and interpretable Digital Twins [56, 33, 102].

Due to this flexibility, there is not a single strategy to generate hybrid models; a wide variety of procedures can be found in literature, depending on the characteristics of every case [58, 134]. In particular, the combination of physics-based models with the increasingly-popular ML techniques has been deeply studied [47].

Hybrid physics-ML models can be formulated using different approaches [130]; the most straightforward is to feed a ML model with the output of a physics-based model to improve the predictions [57, 93]. Other hybrid models propose to replace one or more modules of the physical model with ML models, especially if those are strongly based on hypothesis [92, 59, 27]. A hybrid physics-ML model can also be based on combined predictions of a physical model and a ML model, with careful handling of the corresponding weights [122, 135]. Other types of hybrid models are also developed for inverse modelling or parametrization. Supervised ML has also been accepted for constructing Reduced Order Models. A library of several representations of a physical system under different conditions is generated through expensive knowledge-based models and used to feed a ML-based surrogate model [21, 116, 100].

1.3 Thesis outline

1.3.1 Objectives

The aim of this dissertation is to develop a systematic methodology to tackle numerical simulation models aided by data analysis techniques, allowing for the efficient management of a large volume of information and their appropriate integration in online computational tools.

This thesis focuses on the following aspects:

- The analysis of complex physical processes:
  - Assessment of unit operations from the perspective of computer simulation.
  - Development of different model types (physics-based, data-based and hybrid approaches) and comparison in terms of accuracy, efficiency and simplicity.
- The development of specific tools to promote the performance of hybrid models based on computer simulations at different levels:

  - Data collection: optimizing the design of experiments.
  - Code implementation.
  - ROM enhancement based on clustering identification.

1.3.2 Contents

The outline of the Thesis is as follows:

- In **Chapter 2: Strategies for online process data analysis**, different approaches to model complex physical processes are developed using a rubber mixing process as case of study:
  - In *Chapter 2.1. High-fidelity description of a mixing process*, a simplified knowledge-based model of a Banbury mixer is described and validated using a Computational Fluid Dynamics model.
  - In *Chapter 2.2. Data-driven analysis of mixing process*, process data is used to generate data-based and physically-informed data-driven models, with emphasis in dimensionality reduction methods.
  - In *Chapter 2.3. Application to other processes*, hybrid techniques are applied to other popular material-transformation processes in industry.
- In **Chapter 3: Strategies for building surrogate models**, a set of tools are developed to tackle common problems of the combination of computer simulation and reduced order modelling.
  - In *Chapter 3.1. ROMs in CFD code*, a robust implementation of ROMs in a commercial software is developed using Equations of State as test case.
  - In *Chapter 3.2. ROMs for complex manifolds*, a method to split the domain of the input space based on the manifold features is presented and tested, which significantly increases the accuracy of the model.
  - In *Chapter 3.3. Optimal sequential DoE for ROM building*, a novel selection sampling is introduced to reduce the computational cost of the construction of an efficient ROM.
- In **Conclusions**, the most relevant findings of this work are summarized and discussed.

1.4 Scientific production

The relevant findings derived from the work of this Thesis have been or will be published by the author in the following journals and congresses, sorted in chronological order.

- Conference paper: **Alfaro-Isac C.**, Izquierdo S., Baquedano G. (2019). Data-driven modeling of semi-batch manufacturing: a rubber compounding test case. IEE International Conference on Industrial Informatics (INDIN). Helsiki, Finland. DOI: 10.1109/INDIN41052.2019.8972310.
- Conference paper: **Alfaro-Isac C.**, Viejo I., Izquierdo S. (2018). Data-driven CFD simulation of an industrial semi-batch mixing process. International Conference on Computational Fluid Dynamics (ICCFD). Barcelona, Spain.
- Article: **Alfaro-Isac C.**, Izquierdo S., Sierra-Pallares J. (2020). Reduced-order modelling of equations of state using tensor decomposition for robust, accurate and efficient property calculation in high-pressure fluid flow simulations. The Journal of Supercritical Fluids, vol. 165. DOI: 10.1016/j.supflu.2020.104938.
- Conference paper: **Alfaro-Isac C.**, Izquierdo S., Sierra Pallares J. (2021). Tensor decomposition for discontinuos manifolds: a case study on thermodynamic properties of water. Eccomas congress 2020 & 14th WCCM. Online format.
- Conference paper: **Alfaro-Isac C.**, Juan-Alejandre A., Izquierdo S. (2021). Tensor-decomposition based sequential design of experiments for computer simulation. 10th International Conference on Adaptive Modeling and Simulation (ADMOS). Online format.
- Article: **Alfaro-Isac C.**, Izquierdo S. Process as online rheometer (In preparation).

In addition, the codes written for section 3.1 ROMs in CFD code and 3.3 Novel sequential DoE for ROM building will also be released as supplementary tools in the next version of TWINKLE library.

Chapter 2

Strategies for online process data analysis

In this section, different approaches are explored to model a batch manufacturing process, specifically a rubber compounding. Batch and semi-batch processing are of significant relevance across diverse industrial sectors. Products manufactured through these processes include additives, agrochemicals, dyestuffs, pharmaceuticals, certain polymers or food, among others [12, 13]. In this case, the target application is quality assessment; specifically, a method to predict the final properties of a product manufactured by means of a semi-batch process.

First, a theoretical approach to the system is presented using a set of differential equations. The validation of this model is performed through CFD simulation. The fitting of the experimental data to the equation system allows the prediction of the material properties, but the main focus on this topic is to gain a deep understanding of the physical process and the relation of the measured process variables with the material properties.

Quality tests are usually performed offline in a laboratory. Hence, though this procedure guarantees that test conditions are fully controlled, it is time-consuming, and there is a significant delay in obtaining the results. In consequence a large amount of resources, in terms of both energy and material, is wasted if the product is going off specifications.

In addition, these samples are taken periodically and therefore this method does not account for the variability of the material within the same batch, nor the changes in material properties during the time between the sampling and the availability of the results.

On the other hand, the main advantages of on-process measurements are the reduction on material wastage as well as on manual interventions, as they can be set to provide an automatic response. On-process measurements can be classified into in-line and on-line, whether they are mounted inside the main process stream or are bypassed. The closer to the polymer processing, the closer to the end-product material properties and the shorter the delay is, allowing for real-time process monitoring. However, in-line rheometers present important disadvantages such as interferences from the process to the rheological characterization, and vice versa, and little to no control over the test conditions.

Hence, the main objective is to develop on-process applications for the prediction of the processed material properties. In order to do that, the process data is used to construct and test several purely data-driven and hybrid models. The limitations and advantages in terms of accuracy and requirements are discussed. Finally, two more case studies are presented: a multi-slit extrusion die and a mixing process performed in a Brabender mixer.

2.1 High-fidelity description of a mixing process

Mixing is a complex process due to the diversity of concurrent phenomena. A heterogeneous mixture of components is introduced into a mixer, where the shearing forces produce the dispersion of the components aiming at obtaining a homogeneous material. The shearing stress distribution depends on the mixer geometry, particularly the shape of the blades, and the rheological properties of the compounds. These properties can also change during the process due to particle interaction induced by mechanical agitation (agglomeration, sedimentation, segregation, suspension) or even chemical reactions, which could also affect the energy balance.

Hence, describing the behaviour of this system through numerical techniques is not an easy task. In this work, a simplified theoretical description of mixing is presented through a system of differential equations. In addition, a CFD model is also developed to validate the analytical approach.

The source of the experimental data used for this work is a real industrial-scale manufacturing process and has been provided by a private company, to experience the limitations that arise in actual working conditions. Therefore due to confidentiality reasons, experimental values are not displayed.

2.1.1 Industrial mixing process description

Mixing is one of the most common unit operations in industrial process engineering across many sectors, like chemical, pharmaceutical, minerals, food, plastics, paper and metallurgical industries. The rubber compounding process presented next describes an industrial case that takes place at a factory that manufactures automotive sealing profiles.

The raw rubber is mixed with oils, reinforcing or protective materials, and chemical additives, such as activators, accelerants and sulfurants, in an internal batch mixer at elevated temperature. Moreover, the constituents are served in heterogeneus states: pellets, powder, flakes, liquid.

If the final mixture does not reach an uniform distribution of the diverse elements, a deficient rubber batch is obtained. However, defects of the componding process are usually reported after performing a quality test of a rubber sample in a laboratory, in a later stage of the process, such as rubber extrusion, or in the final product. Therefore, the waste of material and resources is not limited to the current batch or product, but also to the material that is produced while the error was unnoticed.

The complete cycle, from raw material to quality test of the final product, is depicted in the next figure 2.1 :

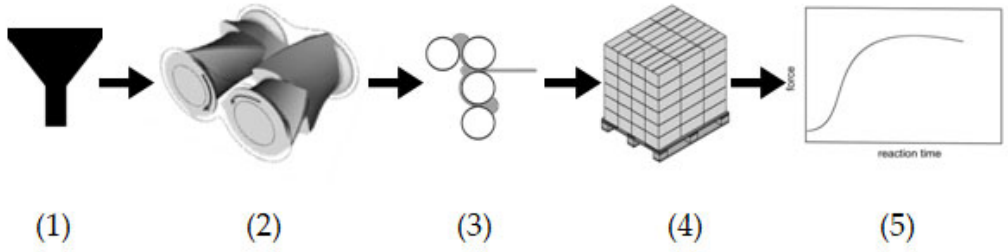


FIGURE 2.1: Main steps of mixing process

The first step, dosification (1), consists of feeding the raw materials into the mixer. Nine different substances, in their required amount respectively, are introduced. The actual weight of each one is also recorded, in case it differs significantly from the ordered weight. The mean weight of each compound is shown in Figure 2.2, where the error bars represent the standard deviation.

The mixing step (2) occurs in a Banbury mixer, characterized by its two counterrotating spiral-shaped rotors. The factory worker states the mixer speed

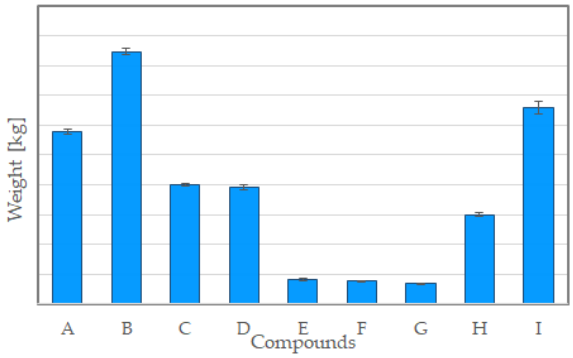


FIGURE 2.2: Process compounds.

and charge/discharge times, and readjust the composition if required. The mixture is passed between calender rolls and sulphurated (3).

Multiple sensors are located inside to provide real-time information of the process, like intensity, pressure and temperature. A supervisory control and data acquisition (SCADA) system collects the mixing process variables every 0.25 seconds from the different sensors. It also storages and organises them in order to present them to the system operators. The process variables recorded during the mixing (intensity, pressure, temperature, rotor speed, gate position) and the calendering (cylinder speed) are depicted in Figure 2.3; in particular, the mean and standard deviation of the available batch data.

The processed material is stored (4) until samples are taken to be tested. Finally, the analysis (5) is carried out offline, in a dedicated laboratory. A material sample is analyzed in a moving die rheometer (MDR) and curing curve parameters are obtained, such as minimum and maximum torque (ML and MH), t5, t10, t50, t90 (time to reach 5%,10%, 50% and 90% of cure), tanD (material damping) and ts1, ts2, ts5 (pre-curing or scorch time). Among them, ML (minimum torque) is selected as the reference quality property, since it is related to the viscosity of the processed material heated to vulcanization temperature. In Figure 2.4, a typical MDR curve is depicted (from Zhang et al. [61]).

Finally, if the processed material fulfils the quality requirements, it is stored in the warehouse until it is used as raw material for other manufacturing processes like injection or extrusion.

Therefore, for each batch, a serie of input variables must be specified and the output variables are recorded along the process, as summarized in the Table 2.1.

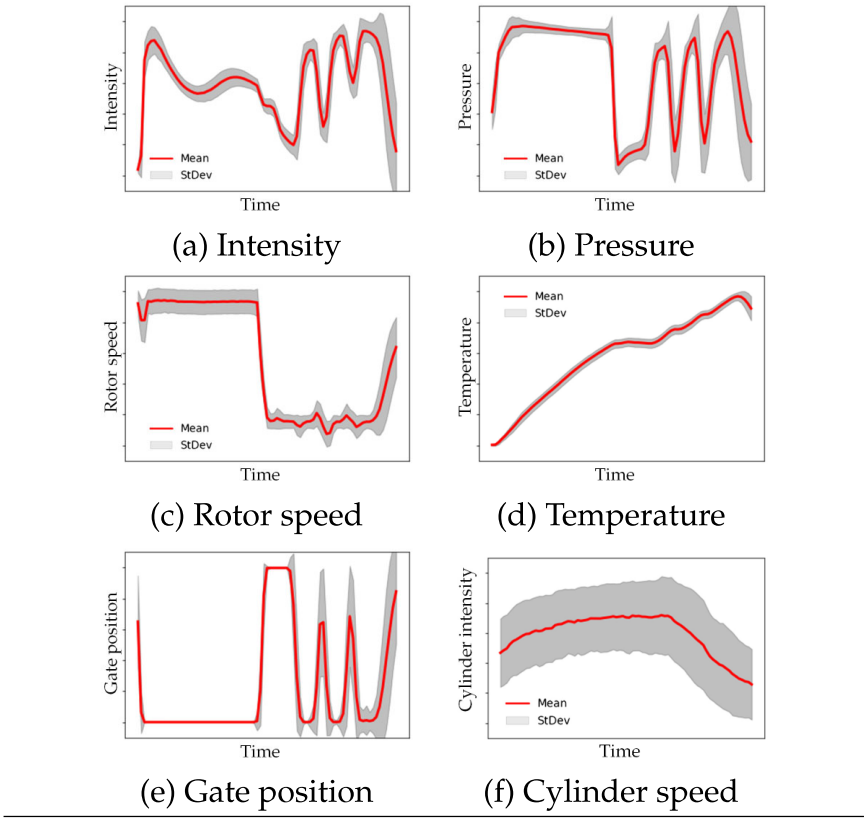


FIGURE 2.3: Collected process variables.

Here the first limitations can be identified. Since the available information is restricted by the sensorization system, some variables that influence the final properties of the material are not collected. For instance, while sulphurization greatly affects rubber vulcanization, the amount of sulphur compound added during the calendering stage is not measured. Another potential issue refers to mixer maintenance: the mixer is not cleaned after every batch and the waste material that is left after one or several cycles on the mixer affects its efficiency, but it can not be determined.

2.1.2 Data cleaning

When the cycle is finished, the data is uploaded and can be accesed through an online server. Before actually using the data, a cleaning stage is necessary to eliminate those cycles where data is noticeable corrupt or incomplete.

Cleaning includes discarding cycles of empty or invalid material tests, as well as empty records of process variables. Some of them also show inconsistent time registers and are consequently removed.

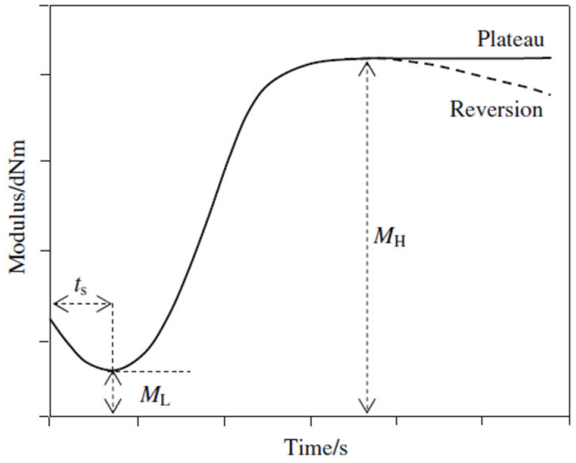


FIGURE 2.4: Representative MDR curing curve.

TABLE 2.1: Input and output variables for each process stage

Step	Input	Output
Dosification	Ordered compounds weight	Served compounds weight
Mixing	Mixer speed Gate position	Intensity Temperature Pressure
Calendering	Cylinder speed	-
Storage	-	-
Analysis	Temperature Time	Curing parameters

Unfortunately, the initial number of available cycles is drastically reduced from 1996 to only 481 after the data cleaning. It must be stressed the importance of the data collection stage to ensure sufficient volume of good quality data to construct reliable models.

2.1.3 Theoretical description

In this section, a simplified physical model of the mixer is described. By applying the preservation laws to the mixing process, this model allows estimating the viscosity of the processed material through a set of ordinary differential equations.

The lack of information regarding the material properties of each component of the mixture prevents a viscosity estimation based on the mixture formulation. Hence, the rheological properties are approximated to a uniform rubber modelled using a power-law equation:

$$\eta = K\dot{\gamma}^{n-1} \tag{2.1}$$

Parameter n represents the flow behaviour index, while parameter K is the flow consistency index of the power-law viscosity model. It varies with temperature and pressure, according to an exponential Arrhenius and exponential relation, respectively, as follows:

$$K = K_p e^{\alpha(\frac{1}{T}-\frac{1}{T_\alpha})+\beta P} \tag{2.2}$$

The indexes α and β designate the viscosity dependence with temperature and pressure.

The actual mixer is represented by two independent chambers with no exchange of forces or fluid through the central plane. Thus, it is considered that the force applied by the impeller of one chamber does not affect the fluid placed in the other.

Each chamber is composed of a single cylindrical cavity of radius R_w and height L. Only the capacity (fluid volume) of the industrial mixer is provided. However, the measurements of chamber radius and length are unknown. The model mixer dimensions are estimated to fit half of the total fluid volume in each chamber. In this approach, it is assumed that there is no free volume inside the mixing chamber. The characteristic, spiral-shaped, blades of the Banbury mixer are simplified and replaced in the model mixer by two radial impellers of radius R_i, height L and negligible thickness.

It is considered that the fluid within the impeller radius rotates at the same speed as the impeller, ω_i. Fluid velocity between the tip of the impeller and the mixer wall is calculated solving the Navier-Stokes equation and imposing no-slip condition on the wall.

$$\omega_i(r) = \begin{cases} \omega_i & \text{if } 0 \leq r \leq R_i \\ \frac{\omega_i R_i R_w}{R_w - R_i}\frac{1}{r} - \frac{\omega_i R_i}{R_w - R_i} & \text{if } R_i < r \leq R_w \end{cases} \tag{2.3}$$

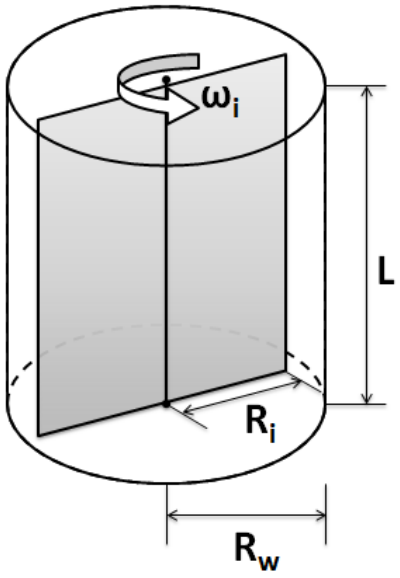


FIGURE 2.5: Model of a chamber of the mixer.

The application of the momentum balance to the mixer model demonstrates that the power system consumption $\dot{W}_{input}$ is due to the power consumed by the rotor impeller $\dot{W}_{rotor}$ and the power dissipated by viscous effects $\dot{W}_{friction}$. Moreover, power is equivalent to the product of voltage V and intensity I:

$$\dot{W}_{input} = VI = \dot{W}_{rotor} + \dot{W}_{friction} \tag{2.4}$$

The power consumed by impeller rotation is derived from the variation of kinetic energy in the mixer:

$$\dot{W}_{rotor} = \frac{\pi L \rho}{6} R_i^2 R_w (2R_i + R_w) \omega_i \frac{d\omega_i}{dt} \tag{2.5}$$

On the other hand, the power lost by viscous dissipation effects is obtained by integrating the shear stress in the control volume.

$$\dot{W}_{friction} = \pi L R_k K \omega_i^{n+1} \tag{2.6}$$

where R_k represents:

$$R_k = R_i^2 + (\frac{R_i}{R_w - R_i})^{n+1} (R_w - R_i)^2 \tag{2.7}$$

The presented system of ordinary differential equations (ODE's) describes the causal relationships between the process variables (intensity, rotor speed, pressure, temperature), the geometrical parameters (mixer length, chamber radius, impeller radius) and the material properties (density, power-law coefficients).

It can be observed that for a single batch process, where the mixer dimensions and rubber characterization are constant in time, the variation of pressure, temperature or rotor speed causes a response in the power consumption of the system in form of intensity change.

$$V\frac{dI}{dt} = \frac{d\dot{W}_{rotor}}{dt} + \frac{d\dot{W}_{friction}}{dt} \tag{2.8}$$

$$\frac{d\dot{W}_{rotor}}{dt} = \frac{\pi L\rho}{6}R_i^2 R_w(2R_i + R_w)(\frac{d\omega_i}{dt})^2 \tag{2.9}$$

$$\frac{d\dot{W}_{friction}}{dt} = \pi LKR_k\omega_i^n((n+1)\frac{d\omega_i}{dt} + \omega_i(\beta\frac{dP}{dt} - \frac{\alpha}{T^2}\frac{dT}{dt})) \tag{2.10}$$

For each cycle, the process data from the physical mixer is collected through the sensorization system and the mixer dimensions are estimated based on the fluid capacity. To validate this approach, the viscosity of the processed material is required. However the only available measurements are the tests performed at the MDR, from which ML can be considered a as relative measure of the viscosity of the non-vulcanized compound. Thus, the viscosity can be related to the experimental ML value, through the rheometer constant, K_rheom.

$$ML = K_{reom}K_{test} = K_{reom}K_p e^{\alpha(\frac{1}{T_{test}} - \frac{1}{T_\alpha}) + \beta P_{test}} \tag{2.11}$$

Experimental data from compounding process is fitted to the ODE's model so the power-law parameters can be optimized and material characterization is completed. Six parameteres should be determined: density (ρ), pre-exponential index (K_p), power-law index (n), temperature (α) and pressure-dependence (β). The reference temperature (T_α) is set to room temperature. However, there is no experimental values of density and thus this parameter can not be validated. Hence, density is selected as approximated standard rubber density value (1000 kg/m^3).

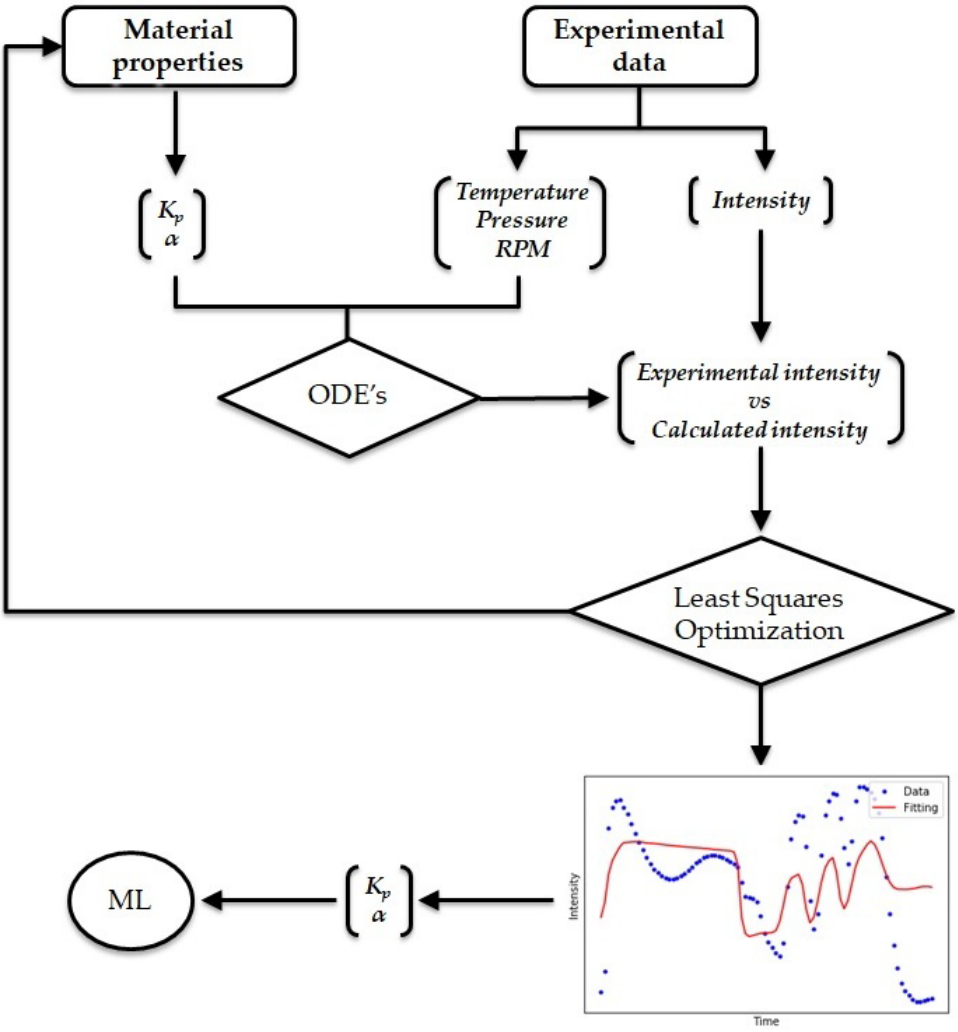


FIGURE 2.6: Algorithm for the calculation of the theoretical model.

Least-squares optimization is applied to minimize the difference among experimental and calculated intensity, which results in the determination of K_p, α, β and n. The characterization from a similar rubber is used to set the initial values, except the pressure-dependence index, which is found in bibliography [49].

2.1.4 Simplified CFD

The flow inside the mixer is simulated using Ansys Polyflow. The aim is to validate the ODE's model by imposing the process temperature, pressure and rotor speed to calculate the intensity in each instant and compare it with the analytical intensity.

The geometry consists of two tangential counter rotating rotors of the same size (R_i, R_w, L) as the ODE's mixer model, as depicted in Figure 2.7.

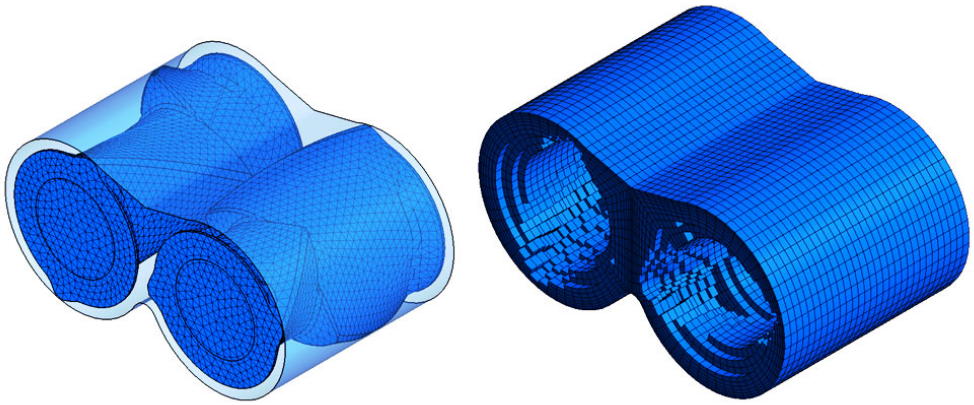


FIGURE 2.7: Mixer mesh for CFD model

The fluid volume is initialized at constant temperature, calculated as mean initial temperature from experimental data, and adiabatic conditions are imposed at mixer walls. Rotor speed is constant and equal to the initial value of mean experimental rotor speed. It is assumed that the fluid sticks to the walls and the rotor surfaces and that the mixer is fully filled. The material properties are set to the mean values of K_p, α, β and n obtained from the optimization of the ODE's system.

Since a simulation of the full mixing cycle is computationally very expensive, only the first five revolutions are calculated. The interpolation scheme is mini-elements for velocity with linear pressure, Picard iterations on viscosity and quadratic elements for temperature. The selected solver is AMF direct solver with the secant iterative process and implicit Euler method for transient integration.

2.1.5 Results

A system of ordinary differential equations (ODE's) that model the flow behaviour during a mixing process is developed. The system is optimized to determine a value of K_p, α, β and n for each data batch. This calculation is performed for the available data of 481 batches and the resulting mean and standard deviation for each parameter are shown in Table 2.2.

The only way to validate the rheological model is to relate it to ML experimental values using Equation 2.11. For each batch, a set of values of n, K_p,

TABLE 2.2: Results of ODE's optimization

Parameter	Mean	St. deviation	Relative st. deviation
K_p	137283.85	19463.67	14.18%
α	106.73	46.55	43.62%
β	5.00E-08	1.32E-23	0%
n	0.2	2.78E-17	0%

α and β are calculated and the flow consistency index K can be determined usign Equation 2.2. Assuming that all the MDR measurements are performed using the same conditions, a linear relationship is expected among the computed viscosity and the experimental ML-value through the rheometer constant $K_{rheometer}$.

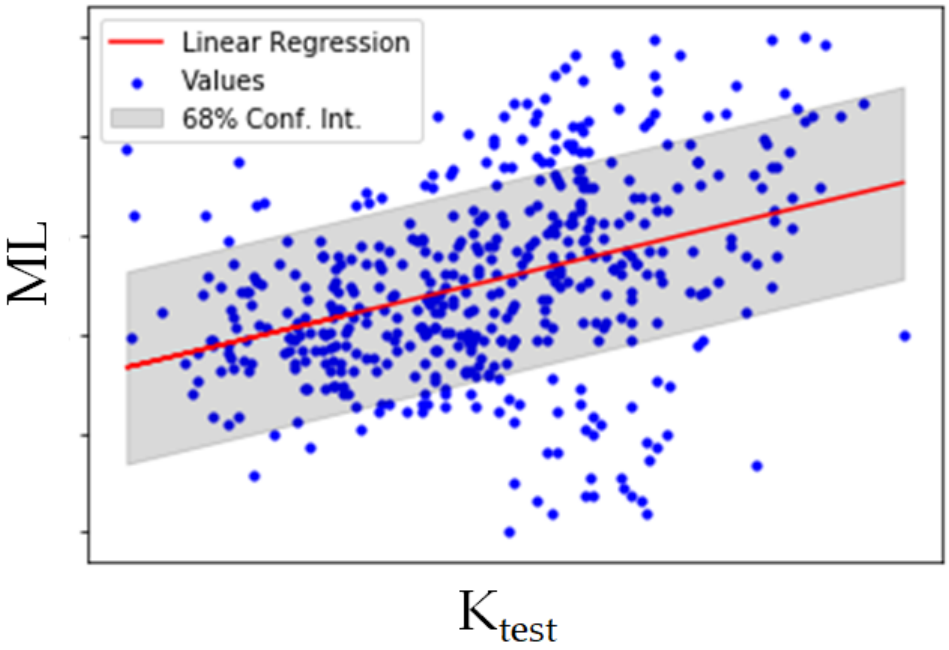


FIGURE 2.8: K_{test} vs ML.

In Figure 2.8, paired values K_{test} - ML for each batch are represented. The linear fit a shows a significant Y-intercept. It is reasonable to consider that the rheometer has a working range and that the offset is related to the lower working limit of the instrument.

On the other hand, the mixing simulation, carried out using the same operational parameters and the mean fitted material model, provides the calculated torque to be compared to the experimental and analytically calculated results. The experimental intensity, the calculated intensity using the

fitted parameters and the numerically-computed intensity using finite element methods are shown in Figure 2.9.

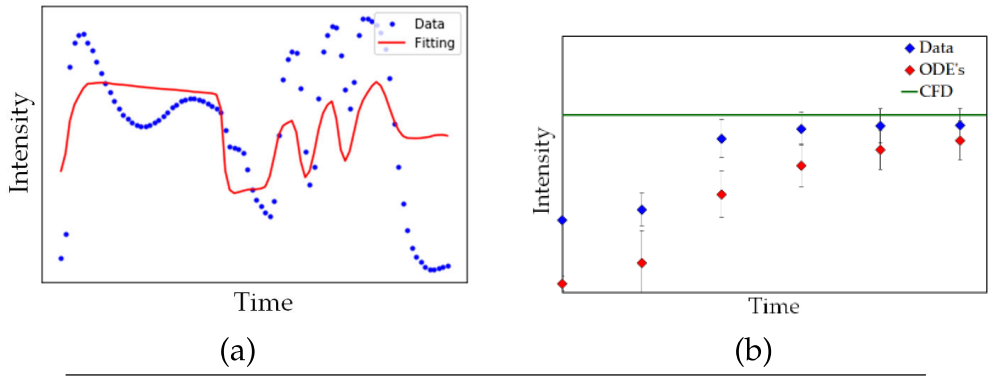


FIGURE 2.9: (a) Experimental and fitted intensity (full cycle); (b) experimental data, analytically-calculated data and CFD simulation results.

In Figure 2.9 (a), the experimental and predicted intensity using the fitted parameters for a sample batch is compared. Considering the numerous hypothesis and simplifications, the predicted intensity is in reasonable aggreement. In Figure 2.9 (b), the intensity calculated through CFD simulation is compared to experimental data and ODE's-determined intensity. The real process experiences a transitory phase, but the same steady value is reached for the current operating conditions.

2.2 Data-driven analysis of mixing process

The development of data-driven models is supported by the intensive sensorization of manufacturing processes that supply large volumes of process data. However, it is rarely the case that a data-based model is constructed directly from process data without any additional transformation.

In particular, due to the dynamic, non-stationary character of batch processes, the data collected from them are usually presented in the form of time-series values, which results in a high-dimensional forecast problem. The so-called curse of dimensionality can hinder the efficiency and reliability of the data-driven model and prevent a proper industrial implementation.

Two approaches are explored to reduce the size of the problem: feature extraction, which is data-driven, or feature selection, which can be guided by

knowledge or data. After the dimensionality reduction step, two multivariate regression methods are also tested and compared regarding their precision and robustness. The combination of a physically-driven feature selection and a machine learning algorithm represents hybrid modelling, while feature extraction or data-based feature selection and multivariate regression are purely data-driven models.

2.2.1 Data preparation

As described in Section 2.1.1 Industrial mixing process description, the sensorization system collects six time-dependent variables: intensity, pressure, rotor speed, cylinder speed, temperature and gate position. For each batch a data matrix is recorded, where the rows represent the timestamps or lectures and the columns show the process variables. To ensure that all data matrices have the same length it is considered that the cycles have the same duration, which corresponds to 79 lectures. Moreover, after the data cleaning (Section 2.1.2), data from 481 cycles are available. Process data variables for all cycles are grouped and organized to build a three-dimensional tensor of shape (lectures, process variables, cycles).

On the other hand, the dosification matrix contains the served weight of each of the nine compounds for each cycle. The output of the system is the quality measurement (ML), which is represented by a single value for every batch. Hence, it is necessary to perform a transformation over the process data to fit it to the dosification and the output dimensions.

The solution is to flatten the process-data tensor in the time-axis to create a matrix where rows correspond to the number of cycles and the columns are variables at a certain lecture. This unfolding technique is common when dealing with time-series data collected from batch processes and can be observed in several works [45, 136, 78, 121]. After appending the served weight of each compound for each cycle, the resulting matrix has 481 cycle rows and 483 features, as depicted in Figure 2.10. It constitutes the input to the data-based model.

Although batch and semi-batch are transient processes, the evolution of their variables is collected at discrete time points; therefore for the building of a data-driven model they can be considered as steady cases where every time instant acts as a new dimension of input data. This observation will be referred to as “batch time”.

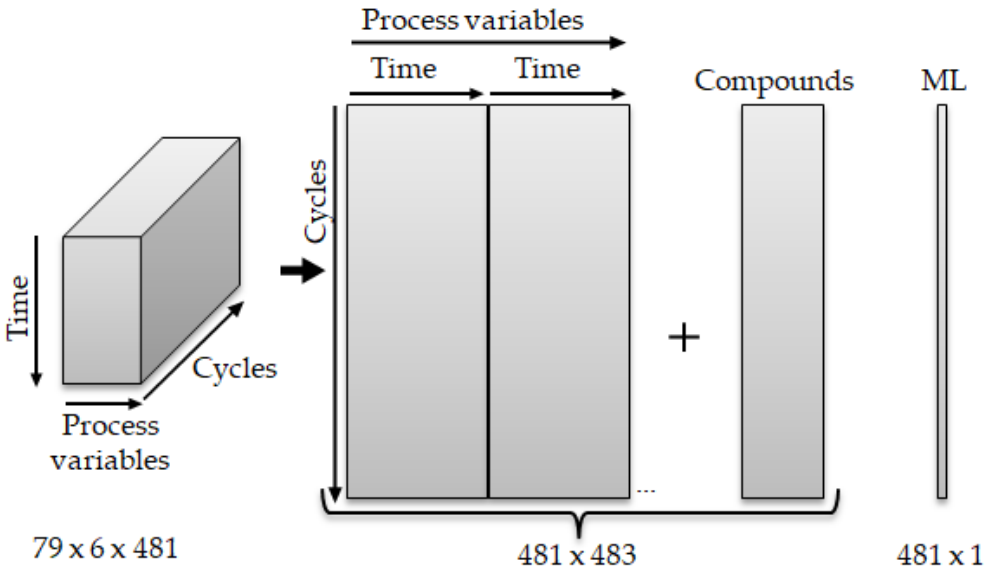


FIGURE 2.10: Input data reconstruction

2.2.2 Data filtering

Manufacturing data collected via sensors is usually affected by random variations or fluctuations that cause interferences in the received signal. Filtration methods aim at reducing the signal noise and smooth time transition. Two popular techniques are tested on the mixer process data: Kalman filter and Gaussian filter.

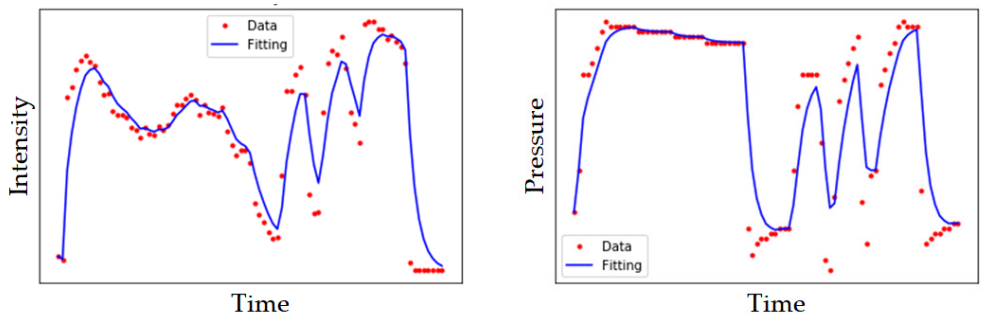


FIGURE 2.11: Kalman filter for (a) intensity and (b) presure.

- Kalman filter

  From an estimation of the process noise variance and the measurements noise variance this technique predicts the next data lecture of a dynamic system. The effect of Kalman filtering on the process intensity and pressure is depicted in Figure 2.11.

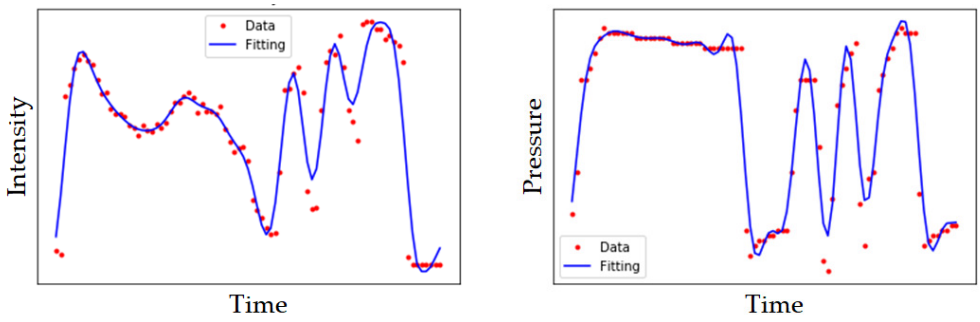


FIGURE 2.12: Gaussian filter for (a) intensity and (b) presure.

- Gaussian filter

  In this case, a standard deviation is provided for the Gaussian kernel, which weights each data lecture according to a Gaussian distribution to calculate the filtered lecture. Gaussian filters smooth time transitions and reduce noise. An example of Gaussian-filtered intensity and pressure is shown in Figure 2.12.

2.2.3 Dimensionality reduction

The dimension of the input data matrix after the transformation described in Section 2.2.1 Data preparation is 483. The number of features is larger than the number of samples, which prevents the data-based model from achieving a good fit and adversely affects the final accuracy of the prediction. In addition, the computational cost of building multivariate regression models raises with increasing dimensions.

Hence, two different approaches are explored to deal with the curse of dimensionality: feature extraction and feature selection [97]. Feature extraction focuses on maintaining the topology of the original data by projecting it into a lower-dimensional space. The most popular methods for feature extraction are based on Principal Component Analysis (PCA) [65]. On the contrary, feature selection preserves the interpretability of the data. Several generalized techniques are found in the literature to apply feature selection to a wide range of problems [46, 67, 20]. An alternative is to select features based on knowledge-guided process-specific criteria [16].

2.2.3.1 Feature extraction

Principal Components analysis is an efficient method to reduce the dimensionality of a dataset while retaining the maximum information [53]. It transforms an initial set of variables, which might be linearly dependant, into a new set of uncorrelated "reduced components". This is achieved by projecting the dataset into a lower-dimensional space in the directions of the data that explain the maximum amount of variance.

In addition, the reduced components are ranked in order of importance. That means that the first component contains the largest variance of the dataset. However, since the resulting reduced components are a linear combination of the initial variables, they lack physical meaning.

The input data for the feature extraction is the resulting dataset after the data transformation performed in Section 2.2.1 Data preparation, which is composed of the time-dependant process variables and the dosification information. In addition, two filtering methods, Kalman and Gaussian filter, have been applied to the process variables. Hence, PCA is calculated on three datasets: the raw input data, Kalman-filtered data and Gaussian-filtered data.

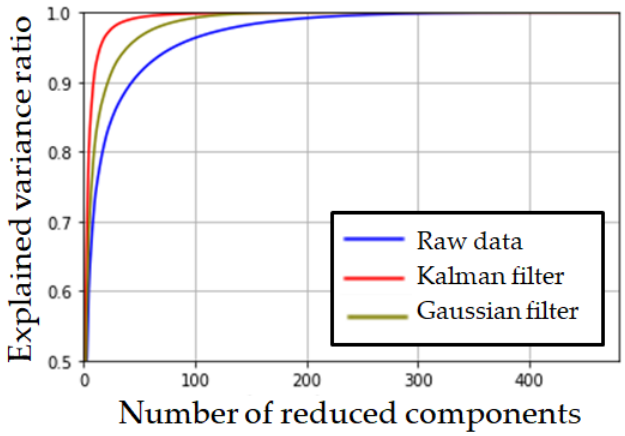


FIGURE 2.13: PCA variance accumulation with increasing reduced dimensions.

The increase of total explained variance with the number of reduced components is shown in Figure 2.14. It can be noted that for an equal number of reduced components, Kalman-filtered data maintains the most information, followed by the Gaussian-filtered set. The optimal number of reduced components is set to 21, which accounts for 96.9% of explained variance in the Kalman-filtered dataset and 82.5% in the raw dataset.

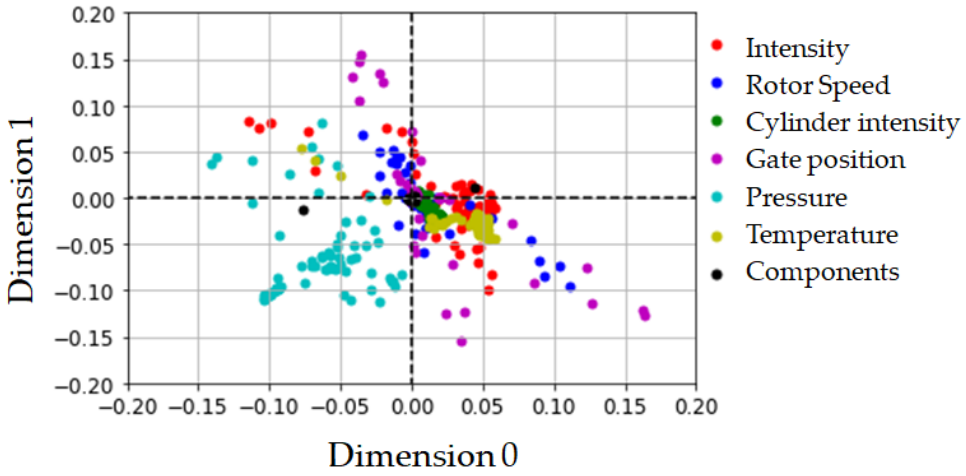


FIGURE 2.14: Weights of the two first dimensions of PCA

In Figure 2.14, the weights of the first two principal components are depicted, coloured by groups of variables. The largest weights correspond to the most relevant variables. The represented distribution is considerable inhomogeneous to raise conclusions, but it can be observed that only two compounds are significantly far from the origin, which corresponds to D and B.

2.2.3.2 Feature selection

Feature selection methods are suited to convert a high dimensional dataset into a reduced dataset that preserves the interpretability of the variables. In the current case of study, the input dataset includes two different types of inputs: the flattened batch time-series and the compounds information. Hence, feature selection is performed separately in each category. For the batch time-dependent variables five proposals for feature selection are presented next:

1. Integral of intensity

   The theoretical model developed in Section 2.1.3 Theoretical description allowed to gain valuable comprehension of the mixing system. One of the main conclusions is that the material properties are intrinsically related to the amount of transferred momentum. This measurement could be represented as the integral value of intensity along the batch time (see Figure 2.15 (a)). Thus, all the features of the batch time-series type could be compressed into a single variable.

2. Variance sorting

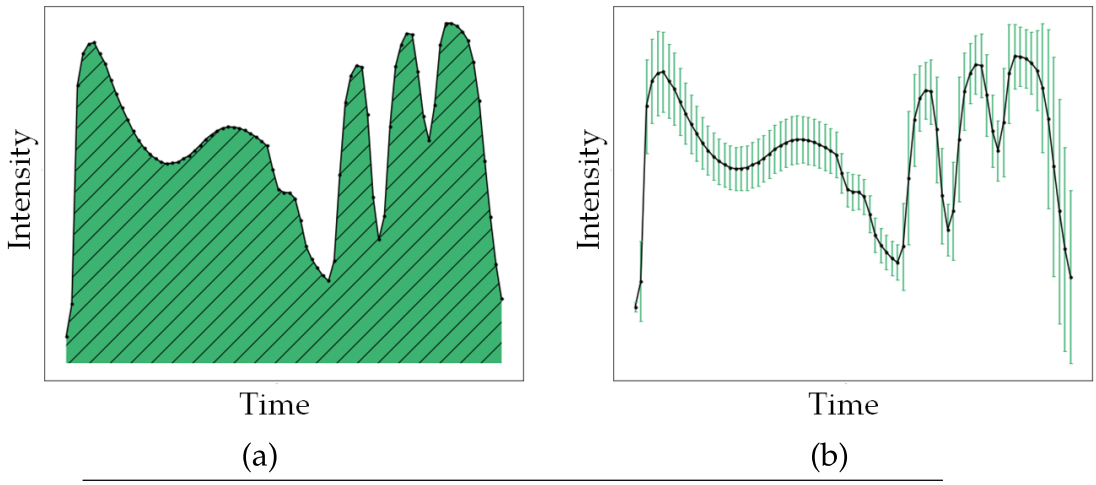


FIGURE 2.15: Feature selection: (a) Integral of intensity and (b) variance sorting.

PCA aims at finding the axis to maximize the variance based on the assumption that a larger variance is equivalent to more information from the system. Instead of projecting the high-dimensional data, in this approach, the lectures of every time-dependent variable are ranked according to their variance (see Figure 2.15 (b)). In addition, the theoretical model inferred that only four process variables are related to the final material properties (intensity, pressure, temperature and rotor speed). Hence, the remaining variables (gate position, cylinder intensity) are not included in this procedure. The dimensionality is reduced to the m variables with the largest variance.

3. Statistical moments

   In this method, the probability distribution function (PDF) is calculated for each batch time series. This PDF can be approximated to a distribution characterized by the main four statistical moments: mean, variance, skewness and kurtosis. This approach is illustrated for the intensity of a batch in Figure 2.16. Thus, the 79 lectures of each batch can be reduced to 4.

4. Backward elimination

   This procedure is based on the importance of the integral of intensity that is previously discussed. Instead of using the final integrated value, this method aims to select the lectures of each variable that can best represent the corresponding integral. To achieve that, the mean of each lecture is first calculated for every process variable. Then, the integral

of the process variable is computed using the 79 timestamps. From there, lectures are progressively removed. The criterion is to discard the timestamps so that the calculation of the integral without it would result in the least variation from the original integral value. As in the case of (b), it only applies to the four specific process variables. Hence, if the user demands that n lectures are required for a proper estimation of the integral, the selected variables would be $4n$. In Figure 2.17 (a), this approach is applied to the mixer intensity, with $n = 5$.

5. Angle method

   The hypothesis in this approach is that the evolution of the system is marked by the time instants where rapid changes occur. Thus, this method aims at collecting the lectures where each process variable shows sudden variations. In practice, this is calculated by determining the angle formed by the current, previous and next lecture for the mean of the timestamps of each process variable. Hence, the n non-consecutive lectures with the largest angle are selected for each batch time series. As in the case of (b) and (d), only four process variables are considered. This approach is depicted in Figure 2.17 (b), using the mixer intensity, with n=5.

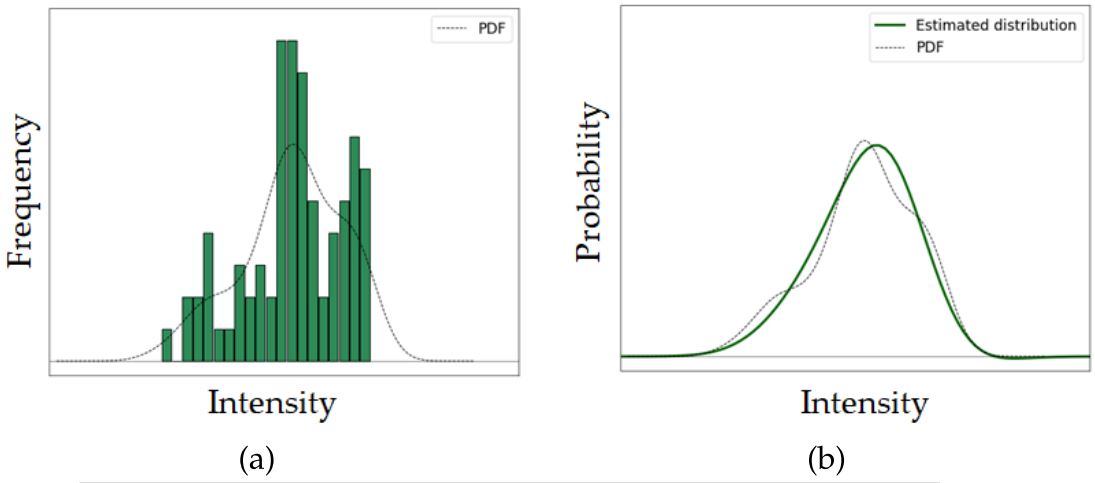


FIGURE 2.16: Feature selection: statistical moments: (a) histogram and PDF; (b) fit of the PDF to a Gaussian distribution.

Regarding compound data, it is not possible to perform a knowledge-based selection since the material properties of each substance are undisclosed. The proposed approach is to select those compounds that most contribute to improving the material quality prediction. To achieve that, sequential feature selection (SFS) [20] is used. The simplest feature selection, which corresponds

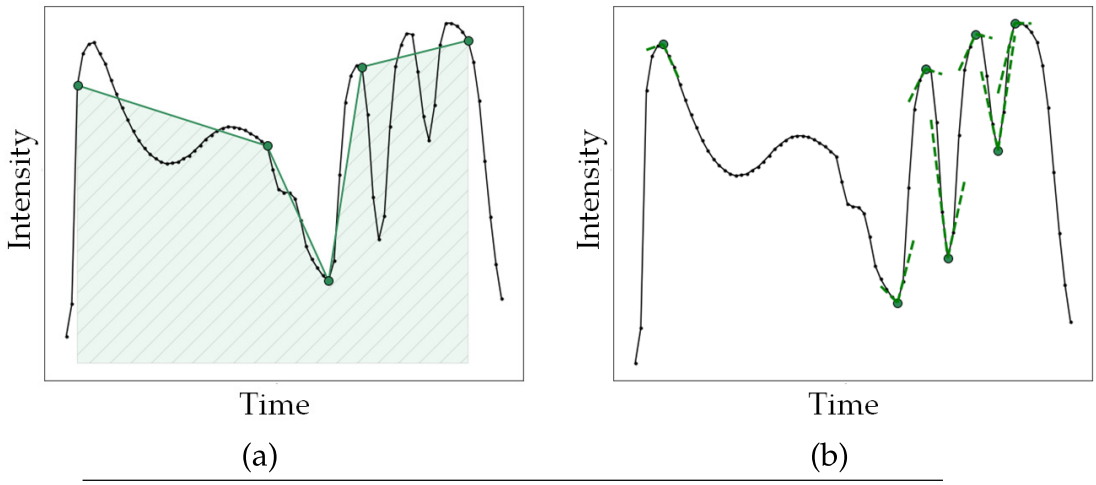


FIGURE 2.17: Feature selection: (a) backward elimination (n=5) and angle method (n=5).

to the integral of intensity method, is used as the zero subset. Linear regression is performed and the mean squared error of the real and predicted values is calculated. Then, each compound is individually added to the zero subset and the corresponding regression and prediction are determined; the feature that scores best is permanently appended as part of the new subset. This selection proceeds iteratively until a performance-sorted list of compounds is obtained.

The results of the SFS of compound data are shown in Figure 2.18. This procedure shows that the largest decrease in MSE is when adding compound D. MSE is improved slightly with the next addition (B) and remains stable for the rest. Therefore, compound D is selected as the most influential regarding the final properties of the material (ML).

From the described feature-selected techniques, five models are developed for batch time-series reduction. For reliable comparison to feature extraction, it is advisable to build models of similar dimensionality. Therefore, the variance sorting method uses the m points with the largest variance across all the process variables with $m = 20$. On the other hand, the backward elimination and angle method restricted the dimensions to n lectures of each variable, where $n = 5$.

To include the contribution of the compound data to the model, SBS was applied and the best compound (D) is included in each one of the five models. The summary of the characteristics of each model is presented in Table 2.3.

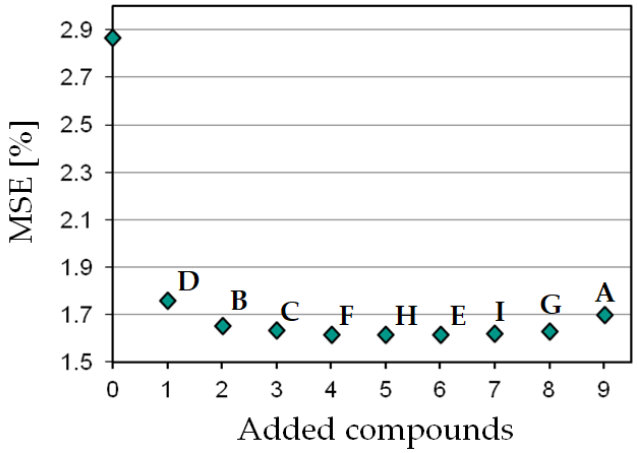


FIGURE 2.18: Feature selection: sequential feature selection of compounds.

TABLE 2.3: Description of feature selection methods

Feature selection	Description	Dimensions
Integral of intensity (II)	Integration	1
Variance sorting (VS)	Top m variance-sorted process lectures	m
Statistical moments (SM)	Statistical moments	4n
Backward elimination (BE)	Top n lectures for every process variable	4n
Angle method (AM)	Top n lectures for every process variable	4n

2.2.4 Multivariate regression

Multivariate regression is applied to PCA-reduced and feature-selected input data aiming at achieving the best predictive model of the output variable (ML). The models are tested using cross-validation [37]. This technique consists of dividing the samples into k groups of the same size and using $(k-1)/k$ sets to train the model (training dataset) and $1/k$ to evaluate it (test dataset). This procedure is repeated k times to subsequently exchange the training and test sets. The global score of the model is obtained by averaging the metric of every iteration. The selected metric to determine the prediction quality of each model is the mean squared error. Two different regression methods are evaluated:

- Gaussian regression (GR), where the kernel is a combination of the Rational Quadratic Kernel plus the White Kernel.

TABLE 2.4: Data-based predictive models: main features

Model	Interpretability	Dimensions
(1) PCA	None	21
(2) II + 1 compound	Momentum balance + Compound analysis	2
(3) VS + 1 compound	Variance analysis + Compound analysis	m + 1 (21)
(4) SM + 1 compound	Statistical moments calculation + Compound analysis	25
(5) BE + 1 compound	Reduced integral calculation (momentum balance) + Compound analysis	4n + 1 (21)
(6) AM + 1 compound	System dynamics + Compound analysis	4n + 1 (21)

- Artificial Neural Network (ANN): The architecture is composed of three dense layers of decreasing output dimensionality. This design follows the general rules of building a feed-forward neural network, which consists of a structure with one hidden layer whose number of nodes is equivalent to the average of the nodes of the input and output layers. The activation function for the first two is softplus and linear for the last one. The selected optimizer is Adam and the mean squared error is chosen as loss function. The ANN is implemented through Keras Python library [60].

2.2.5 Results

For the described models, three quantitative parameters are reported:

- The number of dimensions (d)
- The volume of data required to construct each one (v).
- The accuracy, calculated as the mean squared error (MSE) of real and predicted values of a test data set, for the neural network regression (ANN) and the Gaussian regression (GR).

The results for the five dimensionality-reduction models combined with the two multivariate regression methods are summarized in Table 2.5.

The most accurate model overall is PCA + ANN. Among feature selected models, method (5) Backward elimination shows the best results for both

TABLE 2.5: Data-based predictive models: results

Model	Data volume	Dimensions	ANN MSE (%)	GR MSE (%)
(1) PCA	483	21	0.89	1.64
(2) II + 1 comp	80	2	2.16	1.76
(3) VS + 1 comp	m + 1 (21)	m + 1 (21)	1.58	1.86
(4) SM + 1 comp	475	25	1.72	1.74
(5) BE + 1 comp	4n + 1 (21)	4n + 1 (21)	1.44	1.67
(6) AM + 1 comp	4n + 1 (21)	4n + 1 (21)	1.55	1.79

Gaussian regressor and neural network. It must be noted that the neural network yields a higher accuracy for all the reduction models except for (2) Integral of Intensity. The reason is that since there are only two input dimensions, the intermediate layer size is also set to two, which seems to adversely affect the performance of the method. A comparison of the MSE results is depicted in Figure 2.19.

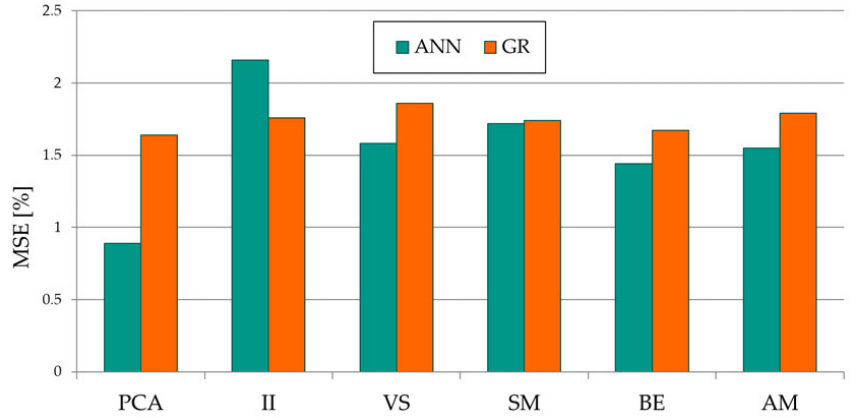


FIGURE 2.19: Results of multivariate regression.

The efficiency of the models can be also evaluated regarding the required data volume and dimensions to obtain the output prediction, see Figure 2.20. It means that despite that the PCA-based method shows the largest accuracy, it requires all the input variables to perform the dimensionality reduction. On the contrary, model (5) Backward elimination shows also a great precision but it only needs 21 variables to directly feed the regression, which represents a significant simplification.

In conclusion, regarding the accuracy of the model, the best result corresponds to the combination of feature extraction (PCA) and artificial neural network (ANN) regression model. However, if interpretability of data is to

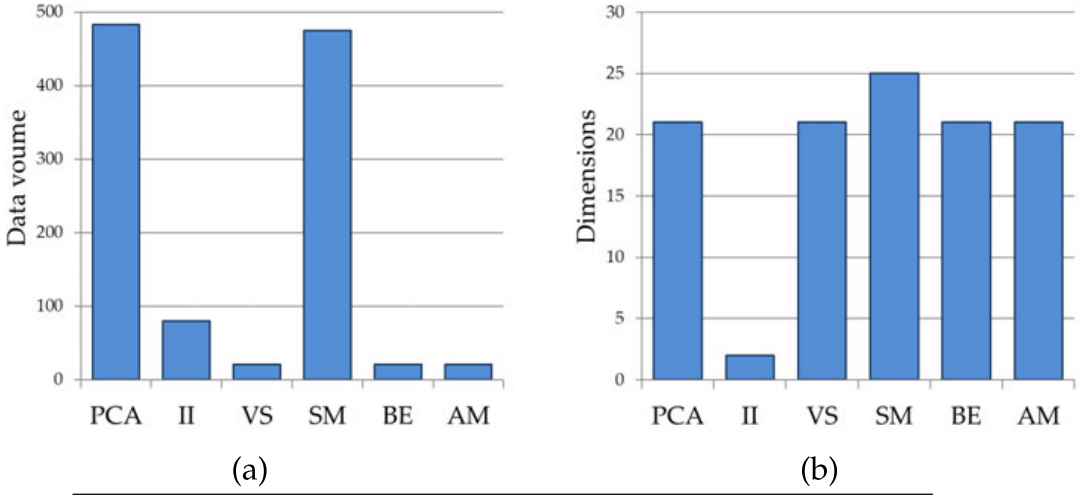


FIGURE 2.20: Requirements of (a) data volumen and (b) dimensions for each feature-selected method.

be preserved, feature-selected is recommended and backward elimination showed the best performance, model (5). From industrial perspective, any of the feature-selected models described in this work could be easily implemented in the process control system.

Finally, considering that these models are designed to be applied to industrial environments, where models can be rebuilt using more data as they are continuously being collected, robustness of both methods for training sets of increasing sizes is a concerning issue.

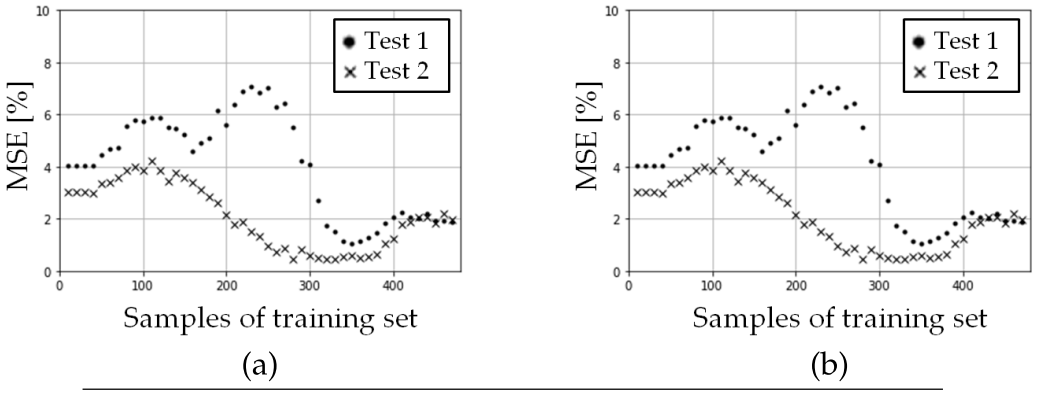


FIGURE 2.21: Increasing training data using (a) ANN and (b) Gaussian regression.

In order to check it, the best scored case among feature-selected models, model (5), is chosen as case study. The procedure is as follows. Eleven samples are blindly selected to make the test set, leaving 470 samples as training set. Two models are built using ANN and Gaussian regression on this training set and tested with the separated test set, calculating the accuracy of

every model through MSE. Then 10 samples are randomly removed from the training set and models are built again using the new smaller training set. This process continues iteratively until reaching a training set of just 10 samples. To check if the differences in the results are influenced not only by the number of samples of the training size but also by the data contained within the sample itself, the procedure is repeated again, so that the randomly-removed samples would be different (the test set remains the same). The results are shown in Figure 2.21. Differences with previous results presented in Table 2.5 are due to not using cross-validation on this test.

It is observed that Gaussian regression is more convenient than neural networks, since the latter do not show stable tendencies when working with different number of dimensions or samples.

2.3 Application to other processes

The rubber compounding case described in the previous section shows several approaches to develop on-process predictive models of material properties. In this section, two additional industrial working cases are presented. The first example is a slit extrusion die, where the viscosity is calculated by integrating the Navier-Stokes equations and assuming a power-law model; next, the rheometry for a compounding process carried out in a Brabender mixer is described through differential analysis.

2.3.1 Multi-slit extrusion die

In this case the aim is to develop an on-line predictive model for the rheological characterization of the extruded material. A slit die is coupled to an extrusion line and a fraction of the polymer flow is derived from the main flow. Two devices are tested: a four-slit and a three-slit designs. Each one has the same lenght (32 mm) but decreasing diameter (3-6mm); for varying rotational speeds at the inlet the mass of extruded rubber is measured.

Assuming a power-law viscosity model for this fluid, as it is developed in [29]:

$$Q = \frac{\pi n}{3n+1}\left(\frac{\Delta P}{2LK}\right)^{1/n} R^{\frac{3n+1}{n}} \tag{2.12}$$

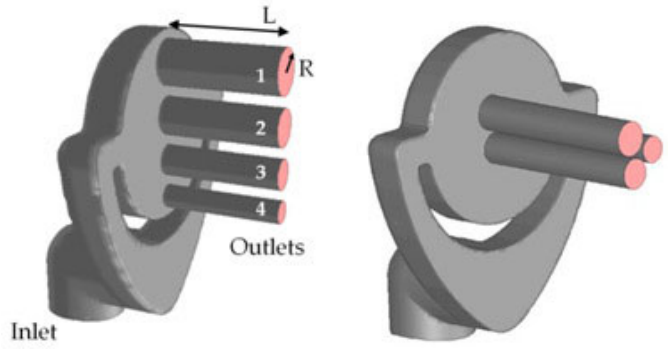


FIGURE 2.22: (a) Four-slit and (b) three-slit extrusion die with slit radious from 3 to 6 mm.

where Q is the volumetric flow; ΔP is the pressure drop between the outlets and the inlet; n and K correspond to the power-law parameters, flow-index and viscous constants, respectively; while L and R are the geometric parameters, length and radius. Therefore, the relation between two channels of different radius and their corresponding volumetric flow is defined as:

$$\left(\frac{Q_{II}}{Q_I}\right)^n = \frac{K_I}{K_{II}}\left(\frac{R_{II}}{R_I}\right)^{3n+1} \tag{2.13}$$

For the four-slit extrusion die, six pairs of channels can be analyzed through equation 2.13; for the three-slit die, three pairs can be compared. Firstly, the volumetric flow ratio is calculated; in order to do so, the density of the fluid must be previously obtained. Assuming that the viscous parameter is constant, $K_I = K_{II}$, the flow-index n can be determined for each pairing according to:

$$n = \frac{1}{\phi - 3} \tag{2.14}$$

where

$$\phi = \frac{log\frac{Q_{II}}{Q_I}}{log\frac{R_{II}}{R_I}} \tag{2.15}$$

For each pair of channels an estimation of n is calculated. To check if the shear rate influences the flow index, for each die the corresponding mean shear rate is computed using the following equation:

$$\dot{\gamma} = \frac{3n+1}{4n}\frac{4Q}{\pi R^3} \tag{2.16}$$

The viscous parameter, K, is determined using equation 2.13 with the n value obtained in the previous step. A system of equations can be constituted by as many statements as considered pairs of channels, where n is constant. For the four-slit die, six equations could be usable but only four are necessary to solve it. For the three-slit die, three equations are obtained. A value of K is obtained for every die and the final value is calculated as the average of the estimated K.

This procedue is applied to two different materials, A and B. The experimental test is repeated five times for material A and six for material B. The nomenclature regarding channel pairings is specified in Table 2.6.

TABLE 2.6: Channel pairing

Relation	**Radius I [m]**	**Radius II [m]**
1	0.006	0.005
2	0.006	0.004
3	0.006	0.003
4	0.005	0.004
5	0.005	0.003
6	0.004	0.003

Material A is tested using the four slit die, while material B is passed through the three slit die. Hence, only relations 1,2 and 4 can be used to power-law parameters determination of material B.

The flow-index n and its corresponding mean shear rate are calculated using equations 2.13 and 2.16 respectively. In Figure 2.23, the mean shear rate is plotted against the flow-index for each relation specified in Table 2.6 to check if the flow-index is influenced by the shear rate.

It is observed that the results of relation 1, which corresponds to the largest slits, are noticeably different in both cases. On the other hand, relations 3 to 6 agree in similar values of flow-index while showing that flow-index is independent of the shear rate. Therefore these data are used to proceed the calculations, discarding relations 1 and 2. The final value is obtained as the average flow index of relations 3 to 6 for material A, and relation 4 for material B.

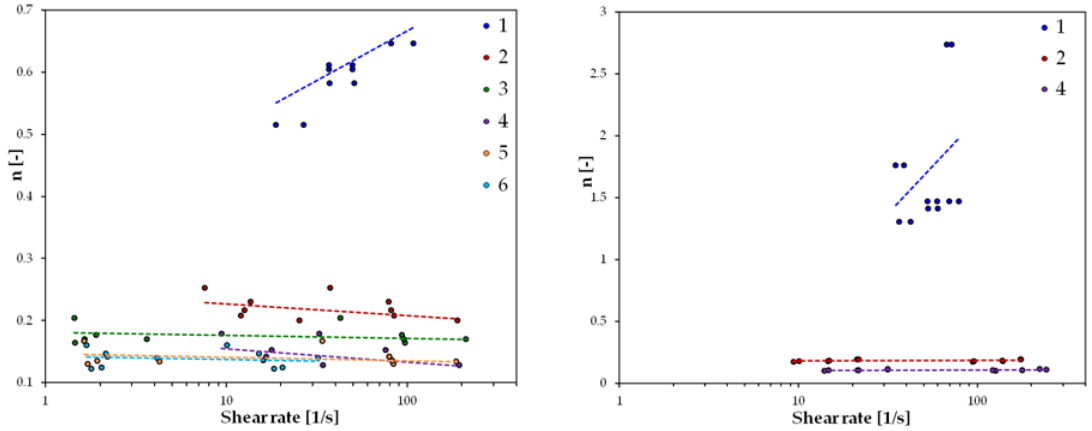


FIGURE 2.23: Mean shear rate vs flow-index for each pairing for material (a) A and (b) B.

Regarding viscous parameter, K, for each relation defined in Table 2.6 a specification of equation 2.13 is defined. In consequence, for each test a system of as many statements as existing relations can be solved. This means, for material A there are five sets of six equations each and for material B there are six sets of three equations. Therefore, a value of K is calculated for every channel and test; the average for each die is shown in Table 4.

TABLE 2.7: Results of extrusion die rheological model

Slit radius [mm]	**K, material A**	**K, material B**
3	153781.0	-
4	148091.3	137424.1
5	145874.8	137380.6
6	166774.6	162851.3

It can be observed that for both materials, the value of K for radius 6 mm differs greatly from the rest of the obtained parameters. This conclusion is in agreement with the previous remarks regarding the flow-index determination, where the largest slits were associated to significant errors. The final values of K are calculated as the mean of the estimated viscous parameter excluding channel of radius 6 mm, and summarized in Table 2.8.

The viscosity flow curve for both materials is depicted in Figure 6, according to the determined power-law parameters.

TABLE 2.8: Rheological characterization of A and B

Material	Flow index, n	Viscous parameter, K
A	0.156	149249.0
B	0.107	137402.4

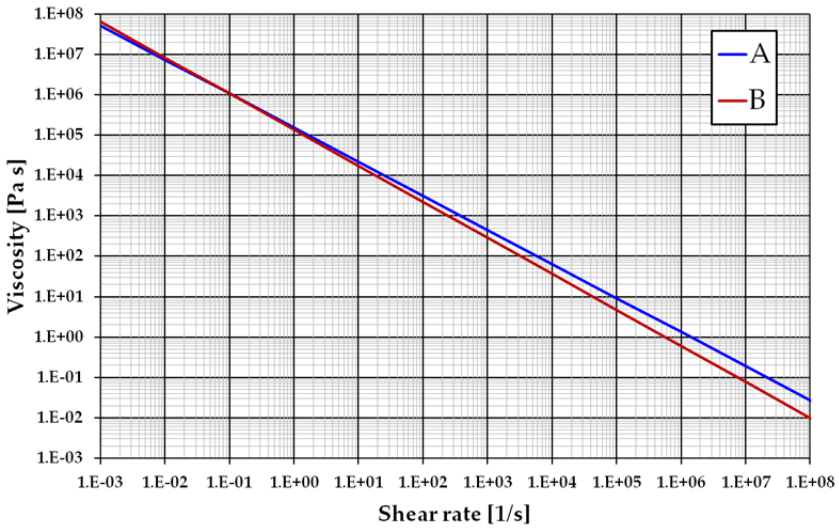


FIGURE 2.24: Viscosity of material A and B.

Thus this methodology allows to determine in-situ the rheological model of the processed material by simply coupling a device to an extruder and analyzing the outflows; nor sophisticated instruments neither complex calculations are required to obtain a precise characterization in short time.

The sources of uncertainties in the measures have the origin in two assumption made. First, the flow is considered uniform, but in the wider slits there are significat flow disturbances at the entrance. In addition, the pressure drop is assumed to be the same for all the channels, but it is actually larger in the slits with small diameters. Hence, the resulting K reported in Table 2.7 moderately increases for the smallest and the largest dies.

Both four and three-slit extrusion dies have been calculated and validated by CFD in an internal report.

2.3.2 Brabender mixer

This case consists of a compounding process, specifically the production of a material composed by a mixture of propylene, additives and wood fibers.

The material is processed in a laboratory-scaled Brabender mixer. The production is non-continuous and the torque is measured and recorded during the mixing process. Thus, a relationship between this torque and the rheology is sought.

In the low-frequency regime, as it is the case, the rheology is shear-dominated, therefore the material can reasonably be modeled using a viscoplastic model. The rheological model chosen to fit the experimental data is the Bird-Carreau [19] constitutive model:

$$\eta = \eta_\infty + (\eta_0 - \eta_\infty)(1 + \lambda^2 \dot{\gamma}^2)^{\frac{n-1}{2}} \tag{2.17}$$

However, in order to perform the fit, the steady-state shear viscosity is needed. Since the experimental set up prevents direct viscosity measurements, the steady shear viscosity as a function of the shear rate is estimated from dynamic data via Cox-Merz relation [19]. According to this empirical rule, the modulus of the complex viscosity can be obtained from oscillatory properties in the low frequency and low shear rate regime as the ratio between the loss modulus and the frequency.

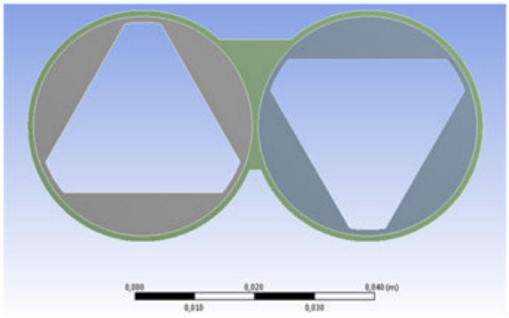
$$\eta(\dot{\gamma})|_{\dot{\gamma}=\omega} = \eta^*(\omega) = \sqrt{\eta'^2(\omega) + \eta''^2(\omega)} \tag{2.18}$$

where η represents the steady-state shear viscosity, η^* is the complex viscosity, η' stands for dynamic viscosity and η'' refers to the elastic contribution. The rheometer used is DMA (cone-plate rheometer).

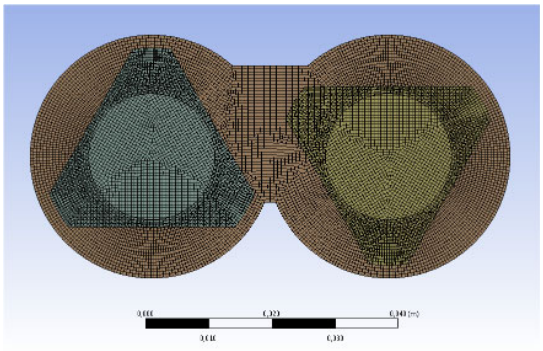
To relate the geometry of the mixing equipment to the material viscosity, the mixing process is simulated using the commercial solver ANSYS PolyFlow. The geometry of the Brabender mixer is accurately reproduced using the CAD software available in the ANSYS WorkBench, namely the Design Modeler application. The simulation set up is specified next. The geometry is 2D and is discretized using mixed quadrilateral and triangular elements for a total number of 13,500 elements. The physical mixer, the geometry and the computational mesh are depicted in Figure 2.25.

Since it is a rotating case, mesh superposition technique (MST) is adopted. The surfaces are meshed separately and the rotor mesh is superimposed to the stator mesh. The cams of the mixer rotate counterclockwise with angular velocity ratio 2/3. The angular velocities of the cams are 90 rpm and 60 rpm

(a)



(b)



(c)

FIGURE 2.25: (a) Brabender mixer; (b) CAD geometry; (c) mesh superposition.

for the left and right cam respectively. The flow is assumed to be fully developed in time; therefore a complete (periodic) rotation of the whole mixer is simulated and then the flow fields are reused for successive rotations. The simulation time is 2 s, which corresponds to 3 rotations of the left cam and 2 of the right one. The mixer is fully filled and the fluid sticks to the walls and to the surfaces of the rotors. Isothermal conditions are imposed.

The experimental data is introduced into the PolyMat module of the PolyFlow software, performing the automatic fitting. The resulting coefficients found for the Bird-Carreau model in this case are: η_∞ = 6500 Pas, η_0 = 80 Pas, λ = 10 Pas and n = 0.55 Pas. The comparison of experimental data

and the Bird-Carreau model is illustrated in Figure 2.26.

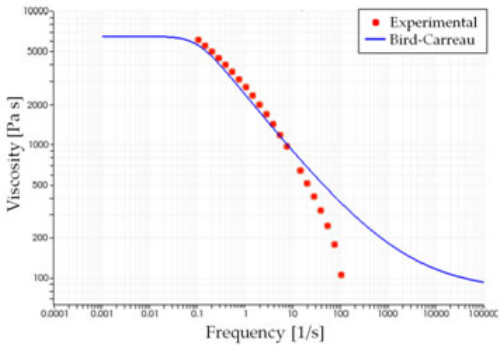


FIGURE 2.26: Experimental data from DMA vs Bird-Carreau model.

The simulation, carried out using the same operational parameters as the experiment and the fitted material model, provides the numerically-calculated torque. In order to validate the results of the simulation, the numerically predicted torque on the rotors is compared with the experimental one in the same conditions. Both cases are reported in Figure 2.27.

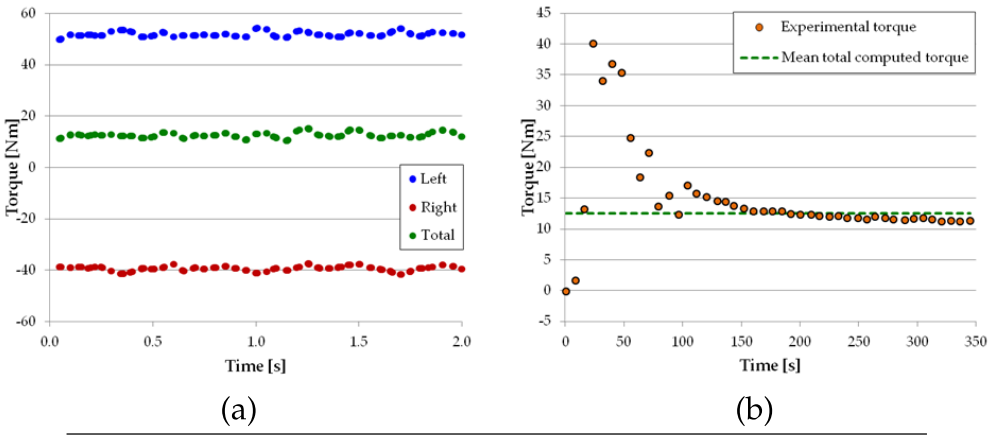


FIGURE 2.27: Torque (a) numerically computed from mixing simulation; (b) experimental vs total computed torque's mean.

In Figure 2.27 (a) the torque corresponding to left and right rotors are shown; the total torque is given by the sum of the torque on the two rotors. In Figure 2.27 (b), the initial transitory phase of experimental torque is due to the inhomogeneous dispersion of the wood fibers at the beginning of the test. After 3 minutes of mixing, the torque reaches a steady value, which corresponds to well-mixed conditions. This value (around 12 Nm) matches the total numerically-computed torque; therefore the experimental data is very well recovered.

The high computational cost of the CFD simulation hinders the applicability of this approach as on-process predictive model. The implementation of this method in an industrial environment could only be possible via reduced order modelling. To achieve that, a full cycle should be simulated for different viscosity parameters and calculate the corresponding torques. A ROM that can accurately correlate the mixer torque and the viscosity represents an efficient tool for on-process in-line prediction of the processed material properties.

2.3.3 Conclusions

In the previous sections, three cases of material processing, one extrusion and two compoundings, have been presented. They all have in common that the applied methodologies seek to obtain the rheological characterization of the processed material, through the determination of the corresponding parameters of the selected model. However, these methologies are adapted to each case depending of individual assesement. The main differences are summarized in Table 2.9.

TABLE 2.9: Description of cases of material processing

Case	Modelling approach	Process location	Onprocess variables
(A) Multi-slit extrusion die	Analytic flow	Online	Rubber flow
(B) Brabender mixer	CFD	Offline	Torque
(C) Banbury mixer	Integral	Inline	Batch time-series

Cases A and C are onprocess, which presents two main advantages: as the characterization is performed in real-time, the measurements are taken under operating conditions and avoids material waste if the product is off-specifications. In addition, case C is located inline; it means that is takes place simultaneously to the process, the production line does not need to stop, no human intervention is required and the bypassed material of an online measurement is saved.

Among the three cases, case C would represent the simplest for deployment, since there is no sensor system which is commonly found in any industrial environment. However, since the characterization relies completely on process data, there is a biggest risk that the recorded data is distorted by noise.

Chapter 3

Strategies for building surrogate models

Reduced order models are the mathematical tool that allows to connect computationally-expensive numerical simulations with the Digital Twin to generate efficient and accurate predictive applications to improve industrial processes. The most direct and simple approach is to use a set of computer simulations to construct a ROM. However, ROMs can also be used to replace a part or parts of a numerical simulation to reduce the computational cost or to simplify that section of the high-fidelity model.

In any of those circumstances, before calculating a ROM is necessary to acquire the data to feed the model by performing a design of experiment. If the ROM is constructed from expensive numerical simulations, an efficient sampling strategy is essential for the optimal usage of the computational resources.

Additionaly, in complex physical systems diverse mechanisms can coexist and the consequence is that the behaviour of the system can change very rapidly. In those situations a single ROM can fail to capture all the relevant dynamics.

In this section these problems are tackled. First, a ROM implementation in a CFD code is developed and tested for introducing equations of state in transcritical flow simulation. Next, a method to perform different ROMs in restricted subdomains of the same system to increase the global accuracy of complex cases is presented. Finally, an efficient sequential design of experiments is proposed and tested.

In this chapter, the computational tools are developed using only TWINKLE

library [137]. TWINKLE performs model reduction through Galerkin projection. Its main advantage is that it is focused in efficient ready-to-use reduced order model calculation and evaluation, and data analysis. In addition, a significant advantage of this library is that the results of the ROM can be compressed into a single text file, which facilitates its implementation in different codes.

3.1 ROMs in CFD code

The implementation of ROMs in CFD calculations to model a part or a subcomponent of the system is useful in different situations, besides the saving of computational resources. If the theoretical base of a concrete phenomena is strongly based on hypothesis, it can be advantageous to use experimental values to construct a ROM that replaces the analytical approach. Another example is when the differences of the scale of the mechanisms are very large: a ROM can represent microscale effects on a macroscale level. ROMs can also be very beneficial in multiphysics simulation; for instance, a computer model with fluid-structure interactions are usually highly demanding but one of the components might be replaced by a ROM that represents its behaviour.

In the first part of this section, the determination of thermodynamic properties in a commercial CFD sofware via TWINKLE is described. To achieve that, several ROMs are calculated for the presented working cases of pure fluids and mixtures and an interface to introduce TWINKLE library as a user-defined function in the CFD software is developed. As a result, a more robust and transferable implementation of the complex equations of state is accomplished. This work was previously published [2], but the ROM interface has not been released at the present time. Next, a TWINKLE implementations in a open-source code is discussed.

Because of the interchangeability capabilities of the TWINKLE library, both the commercial and the opern-source codes that link the ROM file to the CFD software can be adapted to any other simulation. To achieve that, it is only necessary to replace the name of the ROM file and the number of input dimensions to fit the expected dimensions in the computer simulation.

3.1.1 EOS in Ansys Fluent

Equations of state (EOS) play a very important role in chemical engineering technology since thermodynamic fluid properties with low uncertainties are needed for a variety of industrial and scientific applications [114]. In the last twenty years, we have seen the integration of relatively complex EOS in commercial, open-source and in-house computational fluid dynamics (CFD) codes for a variety of applications [90, 14, 111]. This is an important technological challenge in the modelling and simulation of transcritical flows, where the use of equations of state is mandatory due to the enormous variation of thermodynamic properties when the Widom line is crossed [7].

3.1.1.1 Strategies for coupling EOS and CFD codes

A key aspect of the combination of EOS models and CFD codes is the computational implementation of thermodynamic properties calculations to be used by the CFD code. The traditional way is to use an ad hoc implementation of such EOS algorithms for computing density, heat capacity, enthalpy, etc. which is tested versus other already available implementation. Very often, the programming language employed is different in both implementations. Typically, very high-level programming languages are used in an early stage of research of a particular problem -prior to the set-up of the CFD problem- like MATLAB, Python, Visual Basic or Process Simulators like ASPEN PLUS or HYSYS. Normally, in this stage only global mass and energy balances are needed.

When micro-scale information is demanded, a CFD simulation needs to be performed and such high-level implementation needs to be translated to C or C++ code, which are normally the languages used for computing in CFD. This is not a trivial task, and needs to be done carefully because of the appearance of numerical problems and other issues is often unavoidable.

The EOS implementation for thermodynamic properties calculations in a CFD code needs to be robust since every function coded will be required to be called millions of times during a typical calculation. A single failure in a single function can cause the premature end of the CFD simulation with the additional loss of data. Besides, some EOS models are notoriously difficult to implement, such as multiparameter EOS [9] and associating fluid theory EOS [120]. In those cases, it is sometimes cumbersome to write robust, accurate and efficient density computation routines due to the non-linearity of

the problem [8]. Because of that, many researchers advocate the use of cubic EOS, which are easy to work with and a cover very important range of applications [123]. Some open-source and commercial CFD codes have some EOS routines already implemented, such as OpenFOAM [127] and ANSYS Fluent, which has a bridge to connect with the REFPROP library [69].

Existing research recognises the importance of this problem. For the analysis of supercritical mixing layers, Bellan's group wrote some of the first codes integrating EOS and CFD [90, 85, 10]. In those works, the Peng-Robinson EOS [95] with van der Waals mixing rules is implemented in a FORTRAN Direct Numerical Simulation CFD code. The problem under consideration was the binary mixing of nitrogen and heptane, considering heptane is a surrogate of rocket fuel. Later, Meng and Yang developed a preconditioning scheme for the same problem based on partial mass properties and applied it with the SRK EOS [83]. The same authors [82] studied liquid oxygen injection in supercritical hydrogen streams also using a direct simulation with that preconditioning scheme.

In the chemical engineering community, many authors have addressed the problem of implementing EOS and CFD using different approaches. Most of the papers devoted to this issue develop an ad hoc implementation of the EOS. For instance, Sierra-Pallares et al. implemented the Peng-Robinson EOS with different mixing rules for the computational study of different applications of high–pressure technology as hydrothermal flames [110], supercritical antisolvent precipitation [109, 107], nanoparticle synthesis [111, 112] and hydrothermal drilling [108]. In all of the above, ANSYS Fluent software was used, and the EOS was implemented through User Defined Functions (UDF). Raghavan and Ghoniem used the Peng-Robinson EOS along with the Predictive Peng-Robinson 78 (PPR78) approach [52] to perform a direct numerical simulation of water – decane mixing at high-pressure with OpenFOAM software [99].

Other authors have opted for the implementation of wrappers of existing libraries. Vaquerizo and Cocero [124] developed a software bridge to connect the ASPEN PLUS thermodynamic property engine with ANSYS Fluent. Both codes were linked through a complex routine involving Visual Basic, MATLAB and C languages, allowing the complete ASPEN PLUS engine to be used by Fluent. In that paper, the IAPWS [126] and Peng-Robinson EOS

were shown as test cases. Unfortunately, such implementation is not open-source and depends mainly on commercial software. Additionally, the authors do not specify if it is possible its use in a parallel computation. Very recently, Fadiga et al. [34] have developed CoolFOAM, which is a wrapper of CoolProp library for OpenFOAM for compressible fluid flow simulation of single component flows.

Other works deal with the problem in a completely different way, approaching the EOS with a reduced order model (ROM). In this methodology, the different thermodynamic properties to be included in the CFD calculation are pre-computed using available software(commercial, open source or in-house codes) and then approximated by a reduced-order model, which is capable of reproducing the data accurately and fast. Several techniques are available, ranging from the use of deep neural networks to high order polynomial functions. Traxinger and Pfitzner used the Peng-Robinson EOS to train a deep neural network ROM able to reproduce with high accuracy density, enthalpy and heat capacity of nitrogen at high-pressure for a range of pressure and temperature [41] ideal for its use in a simulation of transcritical flow. Cardoso et al. [18] used a polynomial ROM to replicate the results of Sierra-Pallares et al [109] study of supercritical antisolvent precipitation. In this case, density data for the mixture was fitted using polynomials in the range of temperature and pressure for the problem under study.

Thus, ROM can be considered a data-driven approach to the problem of thermodynamic properties calculations. The ROM methodology is promising since it avoids the use of an ad hoc implementation of the EOS in the CFD code, allows for a direct density computation (non-iterative) and the ROM can be generated with state-of-the-art thermodynamics software, without further modification. In addition, the implementation in the CFD code can be unique for different EOS models and very often be much faster than the original EOS code. However, to the authors' knowledge, only the above-referenced papers deal with this problem and for very concrete cases. Thus, the specific objective of this paper is to present a novel methodology for robust, accurate and efficient EOS implementation in CFD codes using ROM for high-pressure, multicomponent transcritical flows. Our idea is based on the use of tensorial networks to approach the EOS data, which is generated using available libraries and later implemented in a CFD code using a universal wrapper. Several strategies can be found when facing tensor decomposition; among them, the most widely applied are Tucker Decomposition

and Canonical Polyadic Decomposition. Both methods are non-intrusive, it means, they are performed on data and the system's equations are not affected [1, 22]. The TWINKLE library used in this work falls into the second category.

3.1.1.2 Tensor decomposition as reduced order model

As stated in the previous section, TWINKLE library is employed for ROM calculation. Since it is a non-intrusive approach, a design of experiments (DOE) must be previously performed.

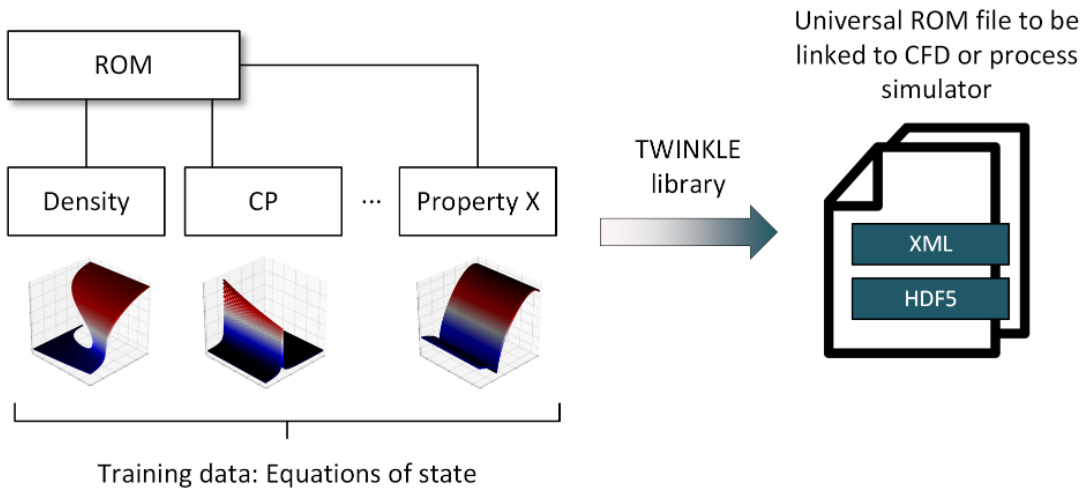


FIGURE 3.1: TWINKLE library to computer simulation

The corresponding EOS is calculated for an interval of pressure and temperature according to the operating conditions defined in the next section if it is a pure fluid case; and component fraction or fractions for multi-component examples. Once the data is obtained, the computational parameters and discretization net is set and the reduced order model is computed using TWINKLE library. The outcome is a single file containing the ROM's results (terms, discretization net, shape functions and weighting coefficients) in a simple format, which allows for universally plugging to any CFD or simulation software (see Figure 3.1).

For each case, the dataset is designed as a full factorial using the same number of levels for each input variable, and evaluated for each output variable. Data on which the decomposition is to be performed is calculated using the Open Source CoolProp library [9] and an in-house thermodynamic code. The generation of thermodynamic data is very straightforward and fast using a

standard workstation, even for a large number of components. Regarding ROM calculation, the computational parameters remain the same for all case studies. Global and term tolerances are set to 0.001 and the maximum number of terms in which decompose the data is 5; the rest are left to default values. On the other hand, the discretization net is specific to each case: the simplest option is to set a uniform net of equally spaced points. To balance ROM's efficiency and precision, it is important not to exceed the minimum number of discretization points that provides a model of sufficient accuracy.

This strategy works well in most cases. However, even if decreasing the step size of the uniform discretization net it does not reach the required precision, it is necessary to adapt the discretization net to the manifold. The subspaces where the model fails are identified and the discretization net is refined on those. For instance, in the cases under study, they correspond to intervals close to critical temperatures and pressures.

3.1.1.3 Implementation in a commercial CFD solver

The calculation of thermodynamic properties in a CFD simulation is carried out because of the need of certain properties when computing with the conservation equations. The energy equation can be formulated in a variety of forms. In this paper we use the ANSYS Fluent © 2019R3 real-gas implementation, that make use of the energy equation based on enthalpy. Following Bird et al [11], the energy equation for multicomponent systems can be written as follows:

$$\rho \frac{DH}{dt} = -(\nabla \cdot \mathbf{q}) - (\tau : \nabla \mathbf{v} + \frac{Dp}{Dt}) \tag{3.1}$$

where

$$\mathbf{q} = -k\nabla T + \sum_{\alpha=1}^{N} \frac{\widehat{H}}{M_\alpha} j_\alpha \tag{3.2}$$

excluding Soret and Dufour effects. In the above equations, ρ is the fluid density, H is the enthaply, $\mathbf{v}$ stands for the velocity vector , $\mathbf{q}$ for the heat flux vector, τ is the viscous stress tensor, $\widehat{H}$ is enthaly per unit mass or partial molar quantity, k is the fluid thermal conductivity and j_α is the mass flux and M_α molecular weight of of component α , respectively.

Enthalpy and temperature are related by the constant pressure heat capacity definition,

$$C_p = (\frac{\partial H}{\partial T})_p \tag{3.3}$$

Thus, expressions for density, enthalpy, heat capacity and partial molar enthalpy are on demand. The last term in Eq. 3.2 is zero for single-component flows and often neglected in multicomponent flows computation. In addition, derivatives of density and enthalpy are also necessary:

$$(\frac{\partial \rho}{\partial T})_{p,y_i}, (\frac{\partial \rho}{\partial p})_{T,y_i}, (\frac{\partial H}{\partial p})_{T,y_i} \tag{3.4}$$

Other quantities are needed only in a post-processing step, such as speed of sound and entropy. The implementation of real-gas thermodynamics in ANSYS Fluent is carried out using the template shown in Figure 3.2. It is worth noting here that transport properties are also included in the template, and need to be properly computed.

```
UDF_EXPORT  RGAS_Functions RealGasFunctionList =
{
  ANYNAME_Setup,                        /* Setup initialize  */
  ANYNAME_density,                      /* density  */
  ANYNAME_enthalpy,                     /* sensible enthalpy */
  ANYNAME_entropy,                      /* entropy  */
  ANYNAME_specific_heat,                /* specific_heat  */
  ANYNAME_mw,                           /* molecular_weight  */
  ANYNAME_speed_of_sound,               /* speed_of_sound  */
  ANYNAME_viscosity,                    /* viscosity  */
  ANYNAME_thermal_conductivity,         /* thermal_conductivity */
  ANYNAME_rho_t,                        /* drho/dT |const p  */
  ANYNAME_rho_p,                        /* drho/dp |const T  */
  ANYNAME_enthalpy_t,                   /* dh/dT |const p  */
  ANYNAME_enthalpy_p                    /* dh/dp |const T  */
  ANYNAME_enthalpy_prime                /* enthalpy */
  };
```

FIGURE 3.2: Template for coding real-gas thermodynamics in ANSYS Fluent

3.1.1.4 Test cases

To show the viability of the method, we have extracted from literature different single-phase flows of different substances (single-component and multicomponent) as shown in Table 1. All of the cases under study correspond to trans-critical or supercritical flows, where the use of an EOS is mandatory.

The maximum property variation is often found close to the critical point, thus providing a good stress test to the methodology proposed.

Nitrogen, heptane and water (cases A, B and C) are selected to test the capability of the method to deal with single-component flows of different molecular complexity. Nitrogen is a non-associating fluid with relatively smooth property variation. On the other hand, water is a self-associating fluid with an enormous property variation, and heptane is considered to have a complexity between both nitrogen and water. On the side of mixtures, nitrogen + heptane is expected to behave smoothly, but CO_2 pairs show a much more nonlinear behavior.

TABLE 3.1: Test cases studied in this work

Case	Operating pressure (bar)	Inlet 1		Inlet 2		Reference
		Fluid	T(K)	Fluid	T(K)	
A	40	N2	105	N2	300	[91]
B	60	Heptane	720	Heptane	300	[50]
C	250	Water	723	Water	298	[108]
D	60	N2	1000	Heptane	600	[90]
E	100	CO2	308	Acetone	308	[28]
F	200	CO2	308	Ethanol	308	
G	120	CO2	308	Acetone + Ethanol (50/50)	308	

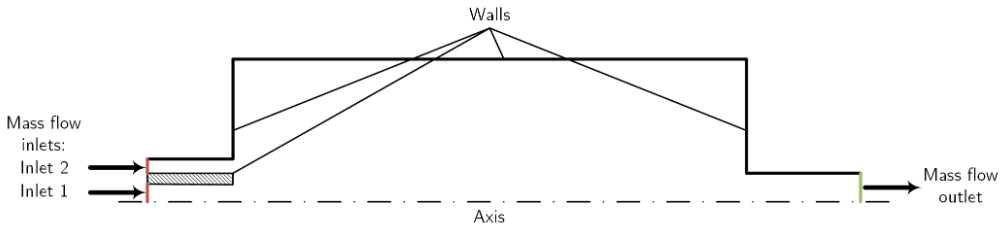


FIGURE 3.3: The computational domain used in this work for simulating the test cases

A sketch of the computational domain and boundary conditions used for simulation is depicted in Figure 3.3. The computational domain used in this work for simulating the test cases. An axis-symmetric domain is employed to ensure fast computation. Mesh is refined close to the mixing point if both inlets 1 and 2.

The calculation of thermodynamic properties using reduced order models considers a different approach depending on the number of components of the test case. For single-component flows, the real gas functions computed via ROM are density, enthalpy, entropy, specific heat, speed of sound, viscosity, thermal conductivity and derivatives of density and enthalpy. In other words, all the quantities listed in the implementation of real-gas thermodynamics shown in Figure 3.2, except molecular weight. For mixtures, only density and enthalpy are modeled using TWINKLE and the heat capacity is obtained through numerical derivation.

TABLE 3.2: Test cases: Training data, discretization type and number of points.

Case	Fluid (s)	Training data	Discretization	Points
A	N2	CoolProp Multi-parameter EOS [6]	Uniform	200
B	Heptane		Uniform	50
C	Water		Adapted	P: 39 T: 41
D	N2 + Heptane	Peng-Robinson EOS with van der Waals mixing rules (BIP =0) [14]	Uniform	200
E	CO2 + Acetone		Uniform	200
F	CO2 + Ethanol		Uniform	200
G	CO2 + Ethanol + Acetone		Uniform	50

A crucial aspect of ROM set-up is the discretization net, especially when variables present strong non-linearities within the defined space. A too coarse discretization will not be able to properly gather these features and the model would reproduce the function with poor precision. On the contrary, an excessively refined net slows both ROM calculation and evaluation, which is disadvantageous in terms of efficiency. For each test case, different sized

uniform nets are set (see Table 3.2). As general procedure, a very coarse discretization is initially designed and tested. Then, the number of discretization points is progressively increased until the corresponding ROM results are accurate enough.

Obviously, this threshold precision is user-defined. For single-component test cases, it is considered that uniform 50-points discretization net provided a high precision ROM for heptane (B), but 200 points are required to reach a similar accuracy in the nitrogen case (A). However, regarding water, even an extremely refined uniform net could not reproduce the sudden property changes around pseudo-boiling temperature and an alternative strategy is adopted.

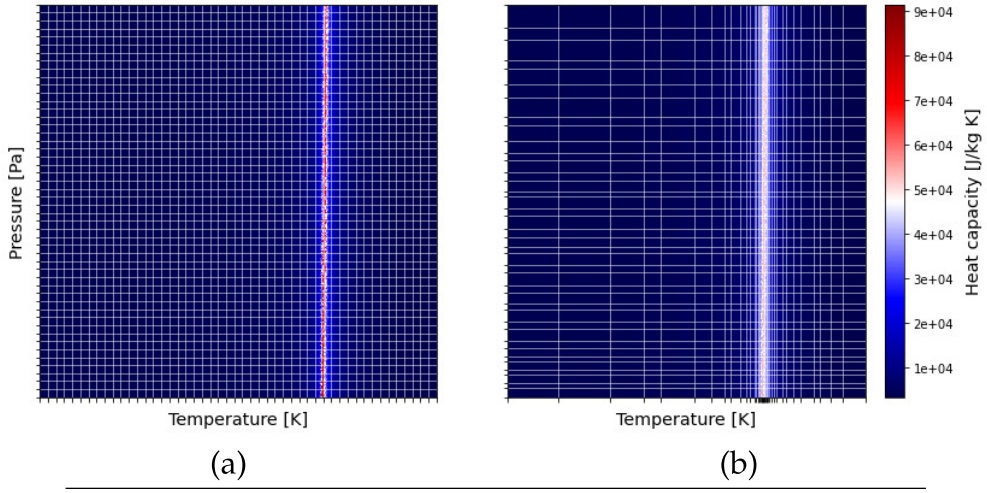


FIGURE 3.4: Water discretization net: (a) exact of 50 points and (b) and adapted

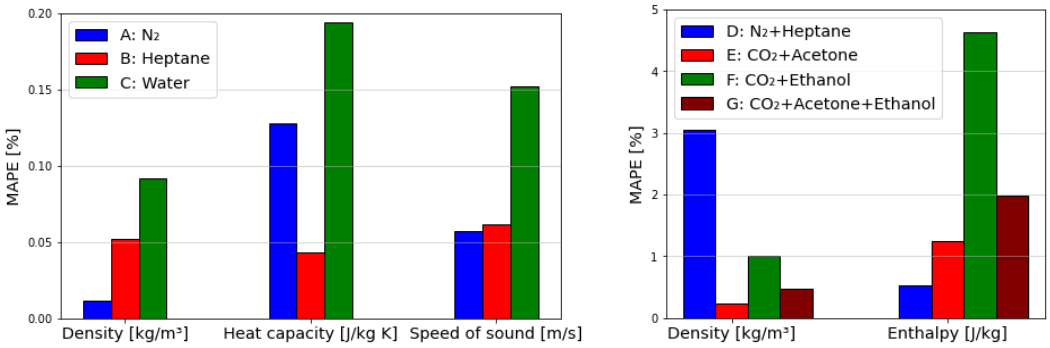
As introduced in the previous section, it consists on increasing the density of discretization points within the intervals of greater variations to better capture the associated non-linearity. As consequence, an non-uniform discretization net refined on those subspaces is obtained, an approach denominated adapted discretization (see Appendix A). A comparison between uniform and adapted discretization net for water heat capacity is illustrated in Figure 3.4.

Referring to mixtures, it has to be considered that the number of dimensions increases by one, including one component fraction (cases D, E, F) in the ROM; or by two, fractions of two components (case G). Since the case complexity increases exponentially with dimensionality, to balance precision and efficiency a lower number of discretization points is preferred in those cases.

3.1.1.5 ROM prediction of thermodynamic properties

ROM calculation is performed for several thermodynamic properties on data acquired through multi-parameter equations based on Helmholtz energy function (HEOS).To assess the prediction accuracy, Mean Absolute Percentage Error (MAPE) is computed between EOS calculated and ROM predicted values.

Selected properties (density, heat-capacity and speed of sound) in single-component test cases are calculated suing HEOS data as training set. The results are summarized in Figure 3.5 (a); while Figure 3.5 (b) shows of all ROM-calculated functions in multi component examples (density and enthalpy).



(a) (b)

FIGURE 3.5: Comparison of test data and ROM predictions for density and heat capacity of water for P=250 bar.

In average, heptane presents the best results, even though the discretization net is coarse. Nitrogen is considered to be the least complex and the discretization net is the most refined, but heat capacity error is higher because the operating pressure in this example is closer to nitrogen's critical pressure.

Covering all variables, the worst scores belong to water, whose modeling represents a challenge due to its characteristics, as described in Test cases. However, all MAPE are below 0.2%. Therefore, in general for single-components substances ROM's prediction accuracy shows an almost perfect fitting. Moreover, across the different thermodynamic properties and single-component test cases this overall trend can also be observed.

In Figure 3.5 (b), on the side of multi-component cases, the discretization net had to be designed using few discretization points to avoid incurring a high computational cost when calculating and evaluating the corresponding

ROM. The analyzed errors are larger than in single-component cases; nevertheless, they remain under 5%. It is remarkable that the most complex case, case (G), which consists of three components, shows an excellent accuracy. This is due to the fact that an increased number of components might act as a smoother of thermodynamic functions.

A direct comparison between the test data and the predicted data for water density and heat capacity for constant pressure P=250 bar and a temperature interval of 500 to 800 K is depicted in Figure 3.6. In each column the ROM is calculated from the same data generated using HEOS but different discretization nets: uniform (exact) of 200 points and adapted. The coefficient of determination (R^2) values are also shown along with each model.

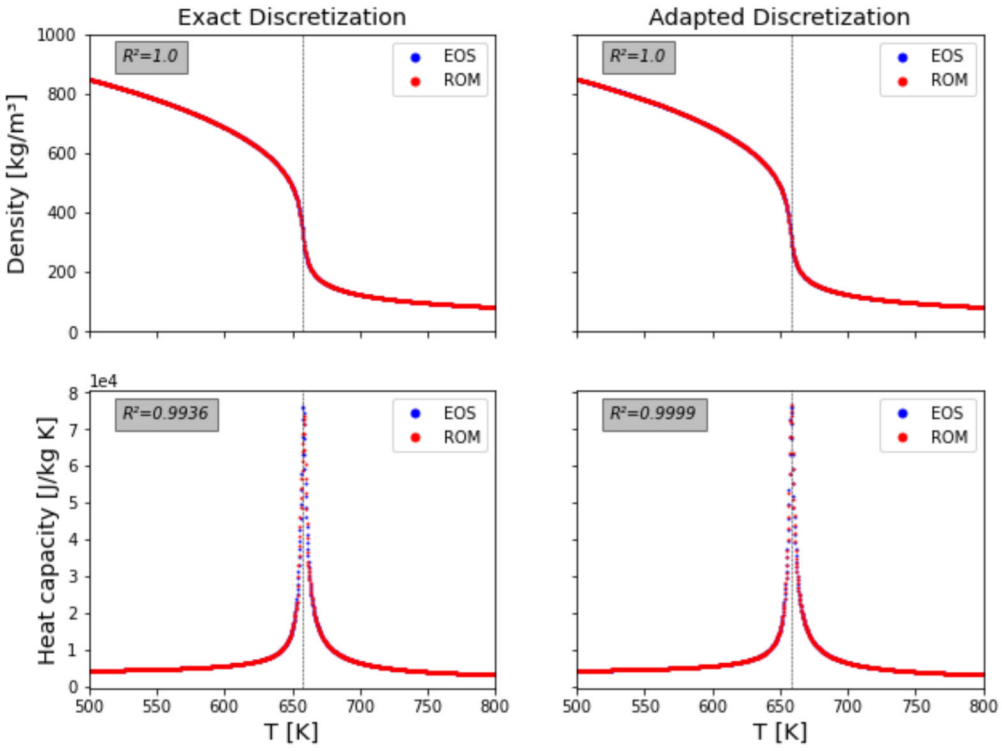


FIGURE 3.6: Comparison of test data and ROM predictions for density and heat capacity of water for P=250 bar.

The density is accurately predicted in all three cases, even around the severe drop, with a R^2 of 1.0. Regarding the heat capacity, it presents a sharp peak at pseudo-boiling temperature that the model built using a uniform discretization net, even a fine one, fails to predict. This is the reason why an alternative net that could include this characteristic feature is developed. The improvement is significant when both net approaches are compared, since the adapted discretization score is almost perfect.

Designing an adapted discretization requires a comprehensive manifold exploration; therefore, it would not a suitable option to apply by default. However, the resulting model benefits from higher precision, simplicity and efficiency, which overcomes the drawbacks in cases as the hereby presented water heat-capacity modelling.

3.1.1.6 CFD simulation of flows with reduced order models for thermodynamic properties

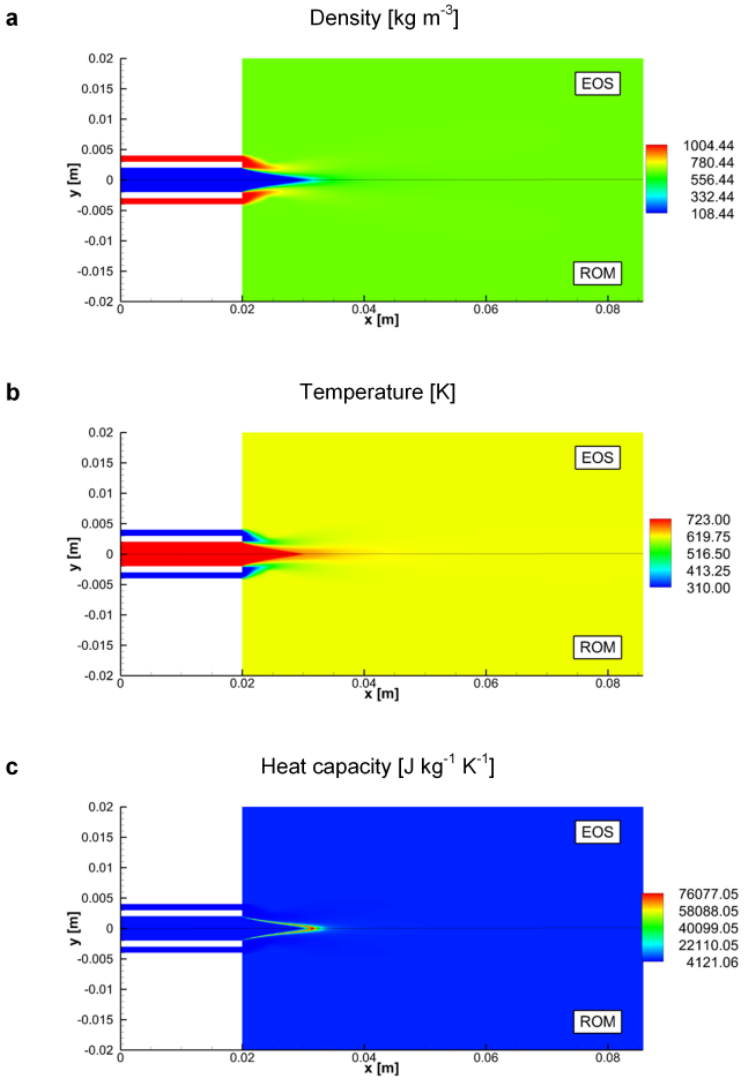


FIGURE 3.7: Comparison of contours for water (Case C); (a) Density, (b) Temperature and (c) Heat capacity. Half - superior image is the result with the full EOS implementation (REFPROP in this case) and half-inferior is the result obtained with the ROM

Once the ROM is properly trained and its accuracy is evaluated, the ROM model is implemented in the commercial CFD solver ANSYS Fluent 19R3 using the template provided in Figure 3.2. In this section, we compare the results obtained for the most representative flows obtained. Regarding the pure component flows, the most interesting to check is water, due to the complexities found in its fitting and explained above.

Figure 3.7 shows the results for a simulation of hot water injection into a subcritical co-flow, similar to those found in [108]. Under these conditions, the jet experiments a massive decrease in temperature, with an enormous gradient and a very steep change in heat capacity. As shown by the figure, the ROM is able to predict extremely well every contour of density, temperature and heat capacity in terms of location and value.

This is better visualized in Figure 3.8, where MAPE in different quantities is represented for the whole domain. Even in this very complex scenario, the deviation is quite low in temperature (less than 1%) which correspond to the peak in heat capacity found at the operating pressure of the simulation.

For the case of multi-component flows, contours of injection of carbon dioxide in a co-flow of ethanol (Case F) are shown in Figure 3.9. Again, the comparison is very satisfactory, and basically the output of both simulations is identical. It is worth to remark here this is a very complex case, in the same range of complexity as water even, due to the extreme non-linear behaviour of the mixture. Here, both fluids are injected at the same temperature which is 313 K, but due to the enormous impact of enthalpy of mixing areas with a heavy cooling and a heavy heating are found in the computational domain. This greatly affects the density distribution, which is correctly predicted by the ROM.

Figure 3.10 shows the MAPE between the EOS and the ROM in mass fraction (a), density (b) and temperature (c). It is clear the ROM is very accurate and almost identical to the result the EOS is giving. Maximum deviation is below 1% for all the quantities of interest. This is an important result, since the coding necessary to use one equation of state or another is always the same. The pairs analysed in Cases E, F and G are representative of the fluids under consideration in the supercritical anti-solvent (SAS) process. With this methodology, it is very easy to tackle a CFD simulation with a complex EOS such as SAFT, PC-SAFT and others with the same codebase, changing only the training data, and obtaining extremely accurate results without worrying about numerical issues. The implementation is very stable, fast and accurate.

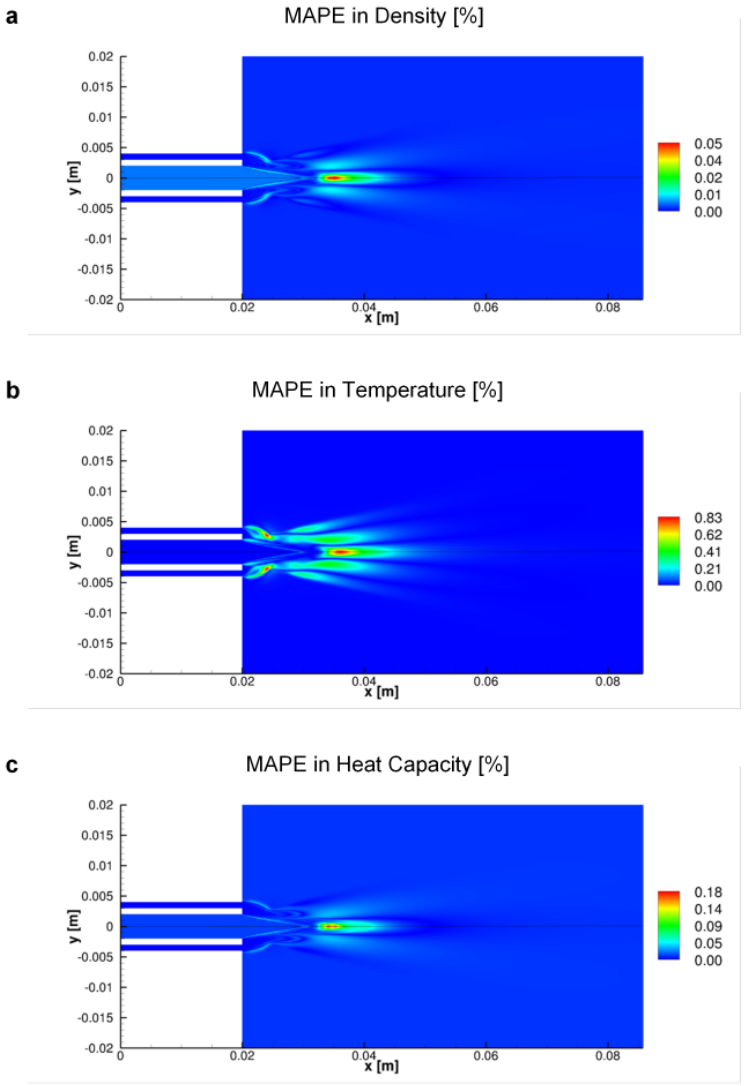


FIGURE 3.8: Mean absolute percentage error (MAPE) between EOS and ROM for case C. (a) MAPE in Density, (b) MAPE in Temperature, (c) MAPE in Heat Capacity

3.1.1.7 Conclusions

It is possible to obtain a robust, fast and accurate ROM of thermodynamic models based on equations of state of arbitrary complexity via the use of a Canonical Polyadic Decomposition based on a Galerkin projection, with a convenient definition of the thermodynamic intervals and training data.

The ROM implementation of thermodynamic properties solves the issue of finding the density from the EOS in each iteration, making a linearization of the non-linear problem in an alternative and extremely accurate way, providing MAPE below 1% for pure fluids and 5% for the mixtures under analysis.

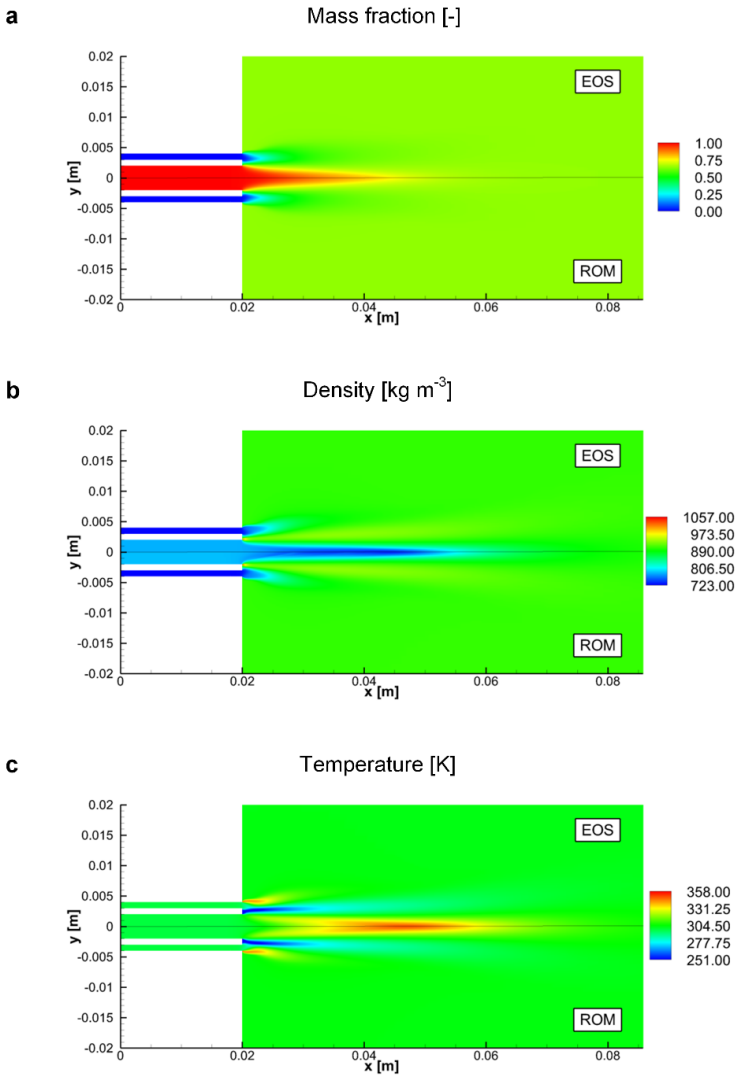


FIGURE 3.9: Comparison of contours for CO2-Ethanol (Case F); (a) Mass fraction of CO2, (b) Density, (c) Temperature. Half - superior image is the result with the full EOS implementation (Peng-Robinson equation with van der Waals mixing rules) and half-inferior is the result obtained with the ROM

When introduced in a commercial CFD code such ANSYS Fluent, ROM interface is satisfactory and universal, making possible the use of different equations of state in different problems with very small changes in the original codebase.Also, it is possible to scale-up the number of components or conditions just changing the training data. This research opens the door to the use of very complex equations of state very easily and efficiently without the need of complete programming of an *ad hoc* interface for each equation of state.

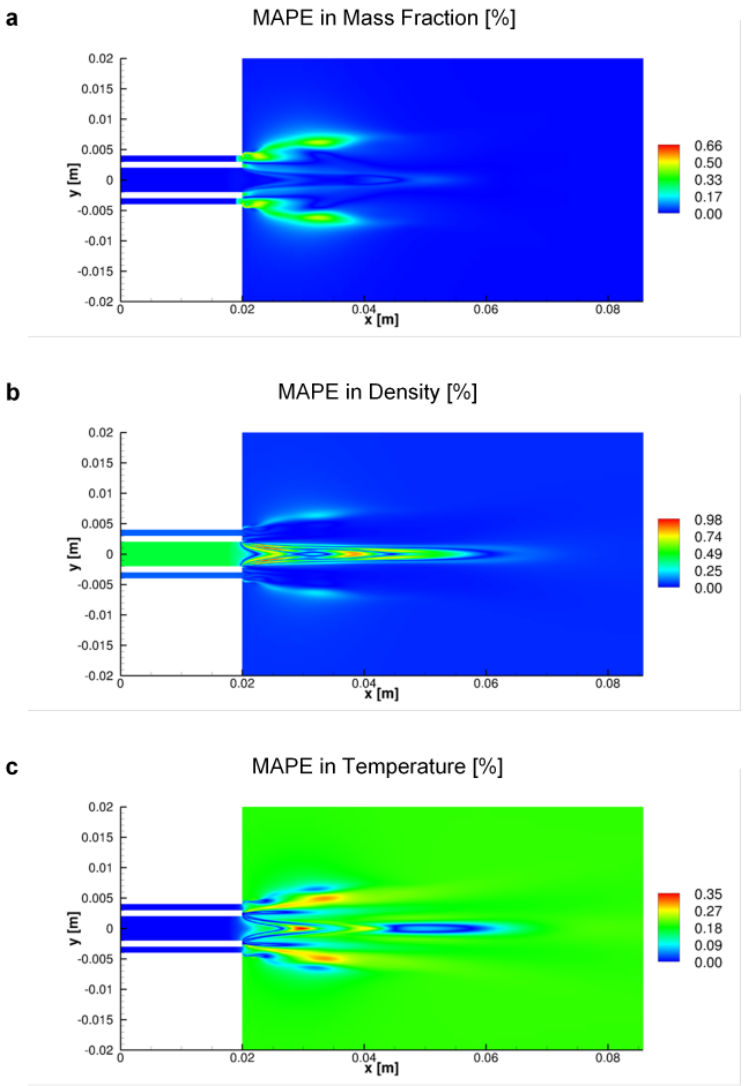


FIGURE 3.10: Mean absolute percentage error (MAPE) between EOS and ROM for case F. (a) MAPE in mass fraction of CO2, (b) MAPE in Density, (c) MAPE in Temperature.

3.1.2 ROM implementation in OpenFOAM

An OpenFOAM [51] implementation of the TWINKLE library has been also recently developed [36]. The case consists of the simulation of a rubber injection process on a mould with micro-textured walls for industrial seal production.

To include the physical texture, which are dimples of 100 μ m of diameter and 30 μ m of height in the computational mesh of the seal model would incur unaffordable computational costs. Hence, a set of microscale CFD simulations of the flow around the dimples are performed under different operating conditions.

From this study, a ROM is constructed that can calculate the equivalent shear stress of a textured wall as a function of the flow velocity, the distance from the wall and the temperature. This ROM is introduced into the macroscale seal model to impose the equivalent wall shear stress that reproduces the effect of the textured mould surface in a flat wall.

While the authorship of the OpenFOAM library that implements the ROM in the CFD simulation belongs to M. García-Camprubi, this author contributed to analysing the data from the microscale simulations and calculating the best ROM from them.

3.2 ROMs for complex manifolds

When building the reduced-order modelling of physical systems, some characteristics of the output manifold topology can prevent the model from achieving high accuracy.

In this section, an unsupervised learning strategy to deal with this type of manifolds unsupervisedly is provided. First, an introduction to the problem is presented, as well as a description of the developed methodology. This procedure is first demonstrated over a case study. Then, this novel methodology is applied to a test case. In both cases, the results are compared to the supervised approach. Finally, the main conclusions are summarized.

3.2.1 Introduction

Tensor decomposition is a reduced-order modelling method that yields explainable models with tuneable accuracy in most cases. However, when the output manifold presents certain particularities, such as discontinuities or very steep gradients, this technique fails at reproducing the manifold topology around it. These manifold characteristics usually correspond to a significant physical change in the system's behaviour.

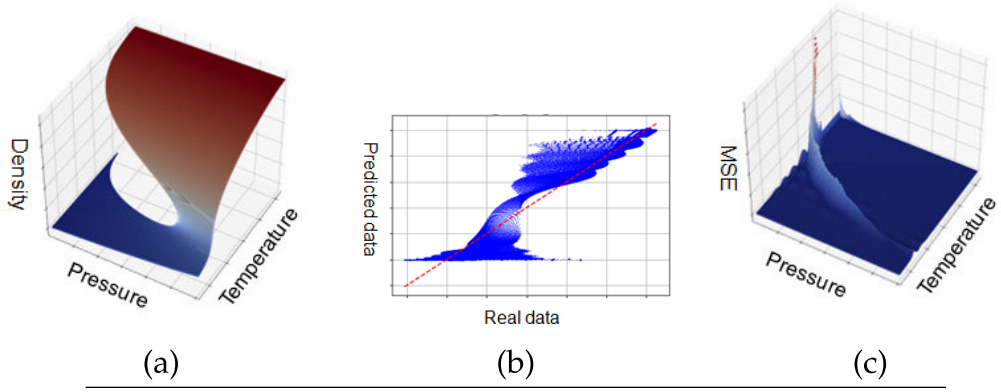


FIGURE 3.11: Example of 3D manifold presenting discontinuity and its model (a) Water density (b) Prediction of water density; (c) mean squared error of water density ROM.

An example is depicted in Figure 3.11 (a), where water density is calculated as a function of pressure and temperature. The response surface shows a large discontinuity across a wide range of the input variables, separating high and low-density values. In this case, the physical interpretation is that this gap represents the gas-liquid phase change. A ROM built from this database fails at predicting the output, mostly underpredicting the higher

density values and overpredicting the lower ones (see 3.11 (b)). The largest deviations, represented as the mean squared error (MSE) in Figure 3.11 (c), mainly trail the manifold discontinuity, proving to be the source of the model's lack of accuracy.

Other examples of manifolds where different dynamics of the system are represented simultaneously in the same output space would be the flow magnitudes under different sonic regimes (supersonic, subsonic); or the properties or quality of a manufactured material under process malfunctions.

Reduced-order models aim at identifying and describing the main mechanisms of a physical system. However, the behaviour of the system can differ significantly depending on the subspace determined by the input variables, and calculating a single ROM for an entire domain can fail at representing the diversity of the system's dynamics, as shown in the previous example. Hence, it is required to develop a specific strategy to deal with manifolds with complex characteristics

A valid strategy is to divide the manifold into two or more clusters to group data whose underlying physics are similar and therefore calculate one ROM for each group that fits each system's response. This manifold division can be performed in a supervised or unsupervised way, before reduced-order modelling. The supervised separation is feasible when there is a clear understanding of the system's behaviour and features, and the data can be classified accordingly.

On the other hand, the unsupervised division must be constructed on the data itself, without a preliminary classification. For that purpose, applying a clusterization method is a suitable approach. It allows identifying the principal mechanisms directly from data in an unsupervised way.

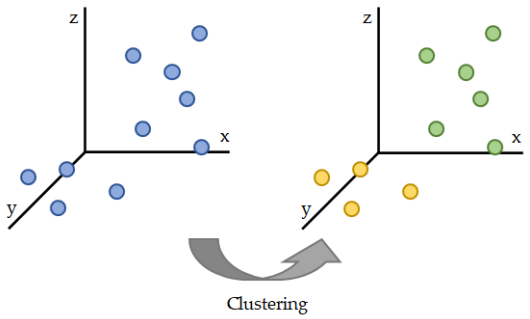


FIGURE 3.12: Clustering as preprocessing for reduced-order model calculation.

This methodology has been successfully developed in several works [54, 4] and applied to numerous fields, such as aerodynamics [115, 129], sensor placement [55], elastic-viscoplastic material behaviour prediction [101] and multiphasic flows [3]. It must be noted that most of them employ distance-based "k-means" algorithm as clustering method.

However, for certain manifold cases, such as the ones presented along this section, the data can be separated based on the knowledge of the physical system but a clusterization method fails at automatically identifying these groups in its primary representation.

The purpose of performing a dimensionality reduction step as preprocessing is to project the current data to a more meaningful space in an unsupervised way. Although a well-tuned clustering algorithm would probably be able to group the data successfully, the advantage of performing a non-supervised separation would be lost.

The novelty of this work is to perform a manifold learning technique over the original data that projects it into a lower dimensionality where the data groups are evident and therefore ready for unsupervised classification via clustering.

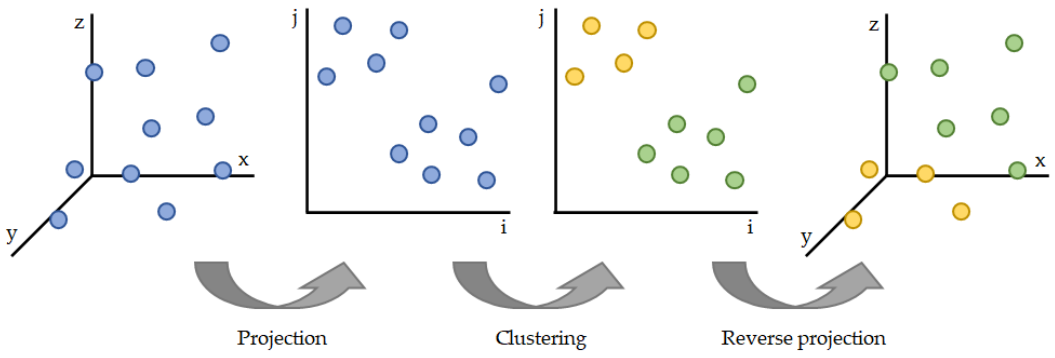


FIGURE 3.13: Dimensionality reduction and clustering.

The workflow is illustrated in Figure 3.13. The data is represented as three-dimensional for better visualization but this methodology can be applied to any n-dimensional space. The data is projected to a lower dimension, where the separation among groups is clear enough so the clusterization can capture it. Finally, the labelled data is depicted in its original representation.

The selected dimensionality reduction technique for this study is t-distributed Stochastic Neighbor Embedding (t-SNE). This method performs a nonlinear dimensionality reduction based on the probability distribution of

the data points. It was originally developed as a visualization tool [79], but its capabilities to infer the implicit structures of high dimensional datasets have proved to be useful in assisting cluster analysis in different fields [70], such as geology [6], [73], chemical physics [119] and genomics [64, 80].

Because t-SNE does not preserve distances, it is not appropriate to use a distance-based clustering algorithm, such as k-means. Therefore DBSCAN, which is density-based, is a more suitable choice. In addition, unlike k-means, the number of clusters does not need to be specified in advance, so DBSCAN is free to recognize as many clusters, and consequently system's dynamics, as it is required.

3.2.2 Case study: proof of concept

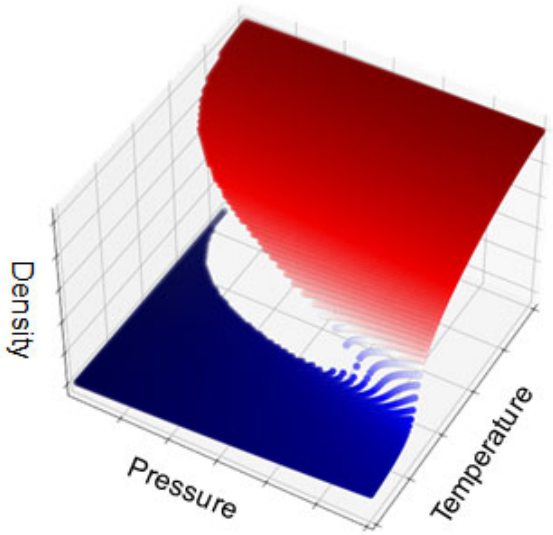


FIGURE 3.14: Water density dataset as function of pressure and temperature, colored by density.

In this section, a case study is presented so the strategies of complex manifold modelling are illustrated. It consists of water density obtained from the corresponding pressure and temperature.

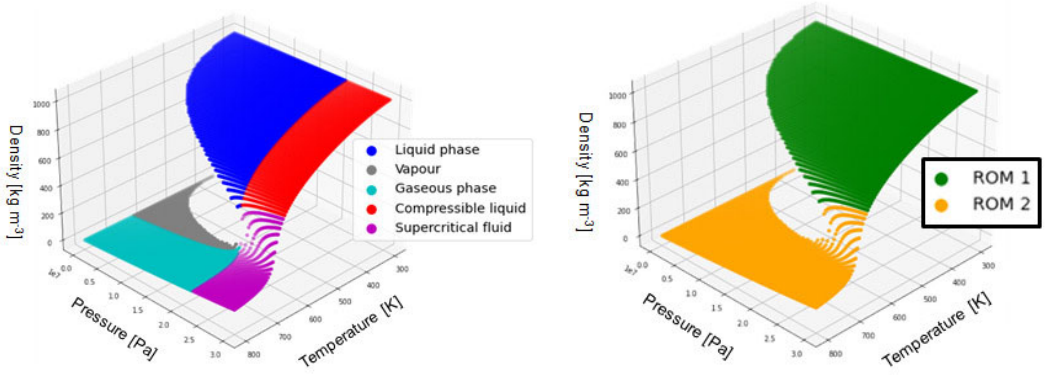
The density data is generated using the Open Source CoolProp library [9], which is based on Helmholtz energy formulations. The dataset is designed as a full factorial of 100 levels. The temperature range is from 298 K to 800 K and the pressure range goes from $1 \cdot 10^7$ Pa to $3 \cdot 10^7$ Pa.

As it can be observed in Figure 3.14, there is an abrupt change in density in a wide range of temperature and pressure. This discontinuity stands for the physical phase change between liquid and solid. It starts in the triple point, where solid, liquid and gas phases coexist in thermodynamic equilibrium (although the solid phase is not represented) and ends in the critical point, above which only one phase exists.

In this example, the discontinuity stands for a simple and well-known physical phenomenon, but there are many cases where it is not easy to locate it by associating it to the process under study, because it is more complex, or several events are happening simultaneously, among other reasons. Therefore, it is advisable to develop a methodology that can identify the manifold complexities with little to no knowledge of the process.

In the following sections, two different strategies are applied to the water density database to split it and obtain two models that can accurately reproduce the manifold represented in Figure 3.14.

3.2.2.1 Knowledge-based manifold division



(a) (b)

FIGURE 3.15: Water density (a) divided by phases; (b) knowledge-based division.

When the manifold complexities correspond to simple, easy-identifiable phenomena, it is possible to split the database according to the process knowledge. In the current case, water density varies abruptly in the liquid-gas phase change, so data can be divided considering the different thermodynamic phases.

Within the given temperature and pressure range, the five thermodynamic phases can be found, as depicted in Figure 3.15 (a). The corresponding physical phase is also provided by the CoolProp library [9].

Because the discontinuity is placed in the limit of vapour and liquid phase, it is not necessary to split the database into five sets but only two. The first set consists of liquid phases, referred to as compressible liquid and liquid phase; while the second one groups gas phases (vapour and gaseous phase) and supercritical phase (see Figure 3.15 (b)). The supercritical phase could be associated with any set because it is located beyond the discontinuity. However, it is decided to proceed with this classification to smooth the density variation at high pressures.

By dividing the domain into two groups of training datasets of their corresponding ROMs it is therefore implied that a previous step to calculate a new sample, this point must be first classified. It means, for a given value of temperature and pressure, it is sorted if it belongs to liquid phases (dataset 1) or gas and supercritical phases (dataset 2), and then calculate the density accordingly.

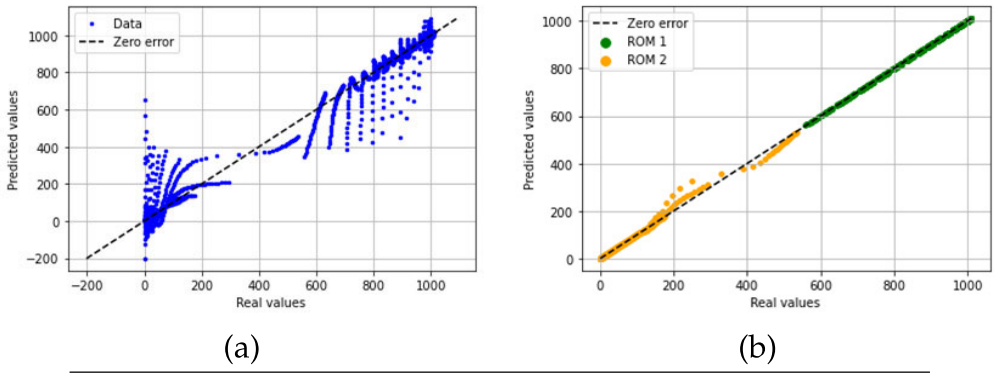


FIGURE 3.16: ROM results (a) single ROM on original dataset; (b) ROM on phase-divided dataset.

Once the dataset is separated, each group is modelled using TWINKLE and two ROMs are obtained. In both cases, TWINKLE parameters are set to five terms and a uniform discretization of 20 points, while the term and global tolerances remain at 0.001. It must be noted that a distinct and customized configuration for each model could be required in more complex cases.

The results of calculating a single ROM over the complete dataset and two ROMs on a phase-divided dataset are compared in Figure 3.16. A test set, different to the training set used to feed the ROMs in each case, is predicted

and the results of the real density values vs the ROM-calculated values are plotted.

It can be observed that when trying to model the water density using only one ROM (Figure 3.16 (a)) for higher values of density that belong to liquid phase, part of them are underpredicted, being mistakingly associated with gas density. On the other hand, for lower densities corresponding to the gas phase, even close to zero, the model often predicts a larger density, such as water-liquid density.

This is due to the difficulty of compressing the behaviour of the boundary of the liquid-gas phase change using a single ROM. The ROM fitting that aims at modelling that discontinuity becomes so ill that even negative values of density are predicted.

However, using two ROMs individually trained with the phased-divided datasets the results show a great improvement (Figure 3.16 (b)). The ROM does not have to capture the liquid-gas limit because it has been already identified. The liquid-phase ROM (ROM 1) perfectly predicts the interval of the highest water density, while the gas and supercritical phase ROM (ROM 2) shows a great fitting with small disturbances that correspond to the supercritical area close to the compressible liquid. The overall mean squared error decreases by 98.85% compared to the single ROM approach.

It is proven in this section that dividing the manifold along the discontinuity results in better modelling. In this case, the physical understanding of the dataset guides the segregation. It is indubitable that grouping the data according to their thermodynamic phase promotes the ROM performance, avoiding the problem of modelling around a large discontinuity that leads to unrealistic results.

3.2.2.2 Automatic manifold division

As shown in the previous section, it was demonstrated that by dividing the manifold using our physical understanding of the data, the model's performance greatly increases. In that case, the knowledge of the system leads the segregation. This section aims to find a method that can perform the same separation automatically.

The proposed methodology consists of two steps. Firstly, a non-linear dimensionality reduction is applied to the dataset. The aim is that the data will be separated by the complexities of the manifold when projected into a

lower dimension. The selected algorithm is t-Distributed Stochastic Neighbor Embedding (t-SNE). As with other manifold learning techniques, it aims at finding non-linear structures in data from data itself but this one is especially suited to extract clustered groups of samples based on local structures. Before the projection, all features are scaled.

Once the data is projected onto a lower dimension using t-SNE, it is required to identify and classify the potential resulting groupings. Among the wide variety of clustering methods, density-based spatial clustering (DBSCAN) is the most suitable, because it is based on point packing and therefore can find non-linear divided clusters. In addition, DBSCAN does not need to previously specify the number of clusters in the data.

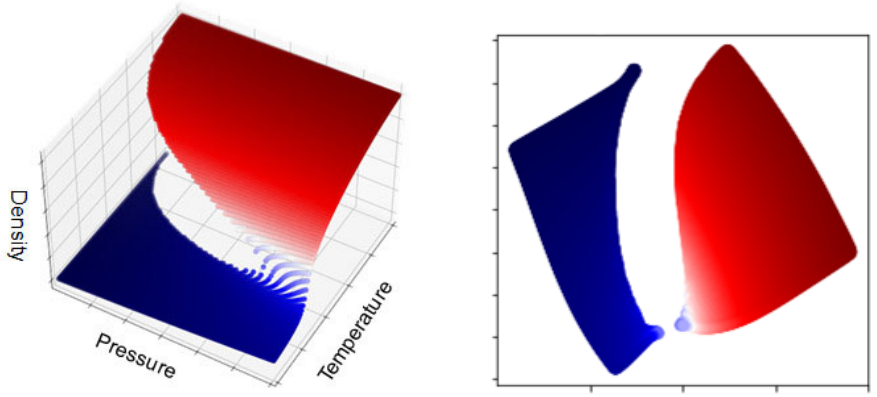
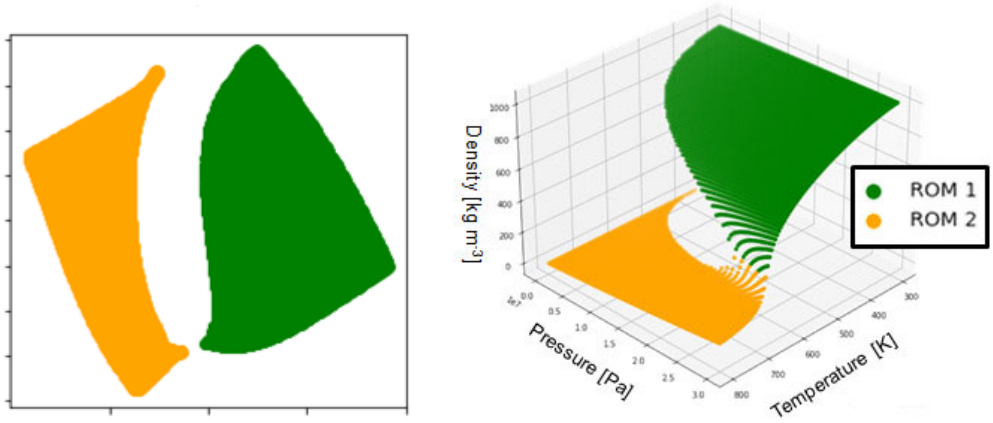


FIGURE 3.17: From original dataset (a) to t-SNE 2D projection (b)

The described procedure is applied to the water density database. The first step is to scale all the features (pressure, temperature, density) to the same range. Next, the three-dimensional scaled data is transformed into a two-dimensional manifold through t-SNE. To do that, the TSNE function from the Scikit-Learn library [94] is used. Two important parameters must be specified: the number of components and perplexity. The number of components is the dimensionality of the output space, set to 2 in this case. The perplexity refers to the number of close neighbours considered for each point; since it is a relatively large dataset, this value is set to 50. The rest of the parameters remain with the default values.

The result of applying t-SNE to the water density data is illustrated in Figure 3.17. The 3D dataset is projected onto 2D space and the algorithm effectively interprets the discontinuity as a data division and expands it to effectively

separate the data into two groups. One of the drawbacks of t-SNE is the high computational cost; in this case, of only three dimensions, the computational time is 43 s.



(a) (b)

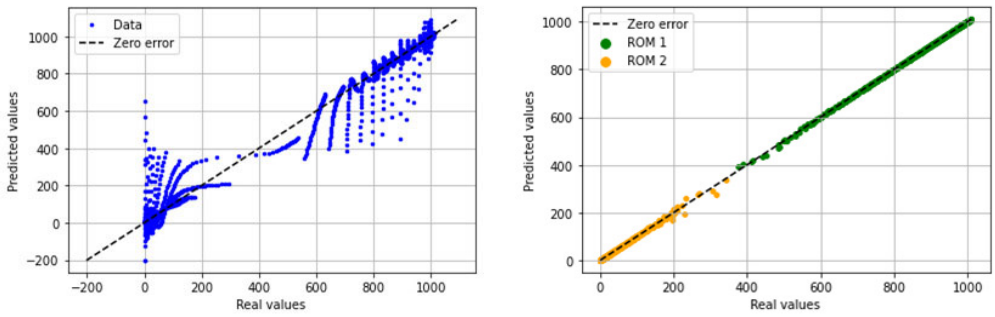
FIGURE 3.18: From clustered 2D projection (a) to clustered 3D original dataset (b).

In the next step, DBSCAN is performed using the homonymous algorithm from Scikit-Learn library [94] to capture the resulting groups after the t-SNE projection. In this case, although the number of clusters is not specified in advance, the algorithm successfully identifies two separated clusters. To reconstruct the original space, each data point is classified and the labels are assigned to the original data. The clustering and final result are depicted in Figure 3.18.

If the clusters obtained from knowledge-guided division (Figure 3.15 (b)) and automatic separation (Figure 3.18 (b)) are compared, it is observed that the division is similar but not identical. At low pressures, the automatic method places the limit in the liquid-gas phase change, therefore in agreement with the phase division. However, at high pressures, on the knowledge-guided method the boundary is set between the supercritical and the compressible liquid phases, the automatic method places the limit at higher temperatures, within the supercritical area.

Finally, TWINKLE is performed on the cluster-divided data using the same configuration as in 3.2.2.1 Knowledge-based manifold division and two models are obtained as a result, one for each dataset.

A test set, different to the training set, is evaluated to compare the predictive capabilities of the single ROM and the automatically-divided ROM's. It must



(a) (b)

FIGURE 3.19: ROM results (a) single ROM on original dataset; (b) ROM on automatically-divided dataset.

be noted that for each new sample to be predicted, it must be classified in one of the datasets (1 or 2) in order to apply the correct ROM. In the knowledge-based approach, it is as simple as finding which phase the new point belongs to. Nevertheless, in this procedure, it is necessary to train a classification algorithm over the labelled data that can assign the corresponding ROM to a new sample.

The results of the test's set prediction using a single ROM over the original dataset and the automatically-divided ROM's over the clustered data set are depicted in Figure 3.19 (a) and (b) respectively. The automatic separation shows a close to perfect fitting between real data and predicted data. The mean squared error decreases by 99.76%, slightly better than in the knowledge-based approach.

It is concluded that an automatic separation can be successfully performed, without any preliminar knowledge of the physics of the system, to obtain efficient ROMs that are trained using each resulting clustered data set. The computational cost of dividing the dataset in the automatic separation is larger than the knowledge-based approach but the results are slightly more accurate.

3.2.3 Test case

In this section, the aim is to model the specific heat capacity of water as a function of pressure and temperature. Both knowledge-guided and automatic approaches, as respectively described in the previous section, are applied and compared.

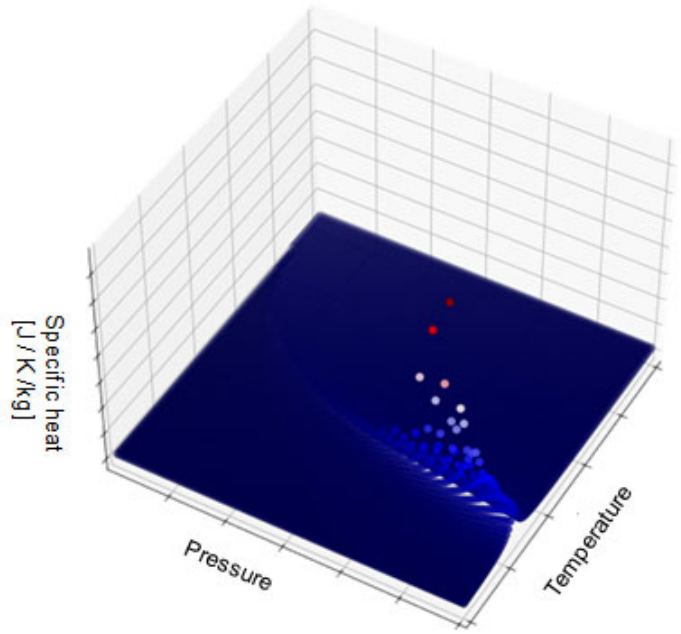
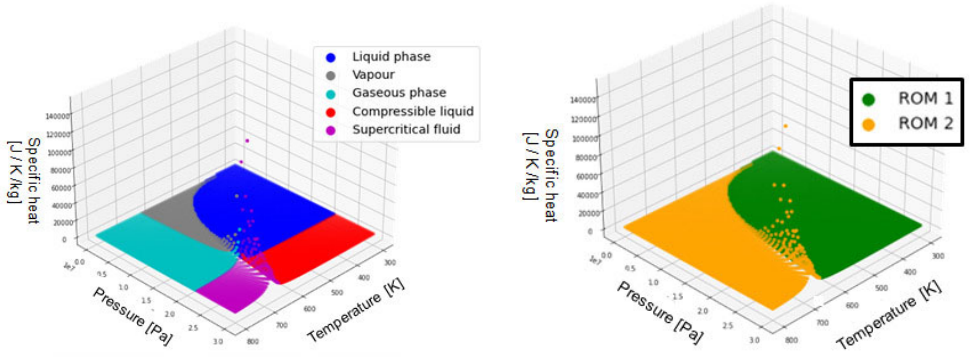


FIGURE 3.20: Water specific-heat dataset as function of pressure and temperature, colored by specific-heat.

The challenge of this case is that this manifold presents a steep peak that corresponds to the critical point, as depicted in Figure 3.20. The dataset is generated through the CoolProp library [9] and using the same specifications as in the previous case study presented in 3.2.2 Case study: proof of concept; it means, a full factorial of 100 levels of temperature in the range of 298 to 800 K and pressure range of $1 \cdot 10^7$ Pa to $3 \cdot 10^7$ Pa.

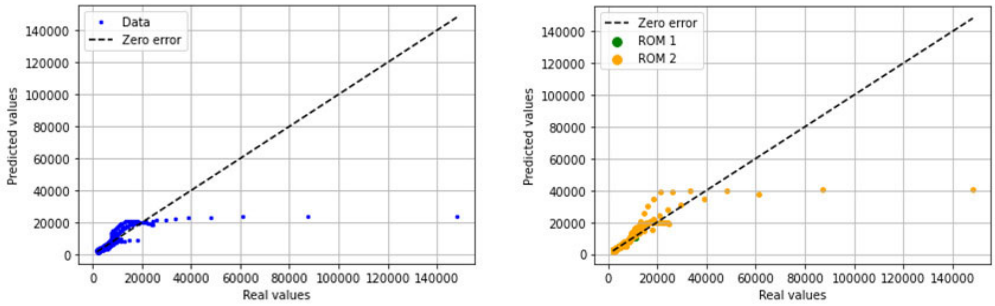


(a) (b)

FIGURE 3.21: (a) Water specific-heat phases (b) Knowledge-based manifold division.

The water specific-heat phases are depicted in Figure 3.21 (a), where it is shown that the critical point is located the intersection between the five

phases. The dataset is divided into two groups according to the phase separation performed in the water density case: the first contains liquid (liquid phase and compressible liquid) phases; the second covers vapour, gaseous and supercritical areas, as shown in Figure 3.21 (b).



(a) (b)

FIGURE 3.22: Specific heat results of (a) single ROM on original dataset; (b) ROM on phase-divided dataset.

Next, TWINKLE runs using the single original data set and the two knowledge-guided datasets to obtain one general model and two phase-divided models. In both cases, the dataset is divided: 80% is used as ROM training data and the remaining 20% as test data. The model is calculated using five terms and a uniform discretization of 20 points for each input variable.

The results of evaluating the test set over the two obtained ROMs are very similar; therefore this phase separation fails at improving the model's accuracy. As observed in Figure 3.22, the higher values of the specific heat are severely underpredicted in both cases. This means that the models are not able to properly capture the large gradients surrounding the critical point.

It is observed in Fig 3.21 (b) that the manifold peak is mostly classified into the second group. In the water density case, this classification was chosen based on the fact that the manifold complexity was located in the liquid-gas phase change. However, in this dataset, where the critical point affects all phases, it could be advantageous to use another classification.

A different knowledge-based division is attempted, where the supercritical phase is separated from its former group. Consequently, three different clusters are obtained: liquid and compressible liquid (1); vapour and gas (2) and supercritical phase (3).

The ROMs are calculated on the three clusters respectively using the same specifications as in the previous case. The results of evaluating the separated

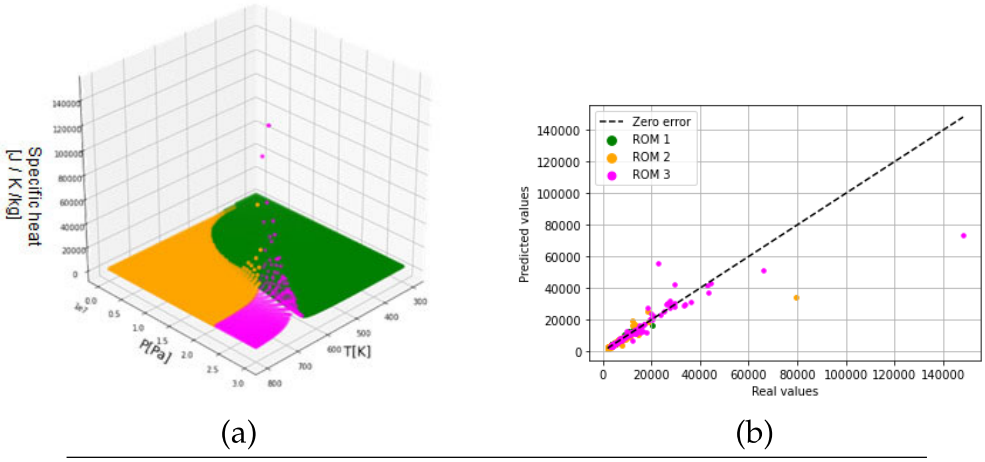


FIGURE 3.23: (a) Phase-based division in three clusters (b) ROM results.

test set show a significant improvement compared to the two-group phase-division and the global R^2 reaches 87%. Figure 3.23 illustrates the new phase-guided division and the corresponding results.

Isolating the supercritical phase allows better modelling of that area, although a relevant underprediction of the highest values of specific heat is still observed in groups 2 and 3. Although the final results are moderately accurate, it is obvious that strong supervision is required. For each case, it is necessary to evaluate the manifold to identify their complexities and adapt the knowledge-based division in consequence.

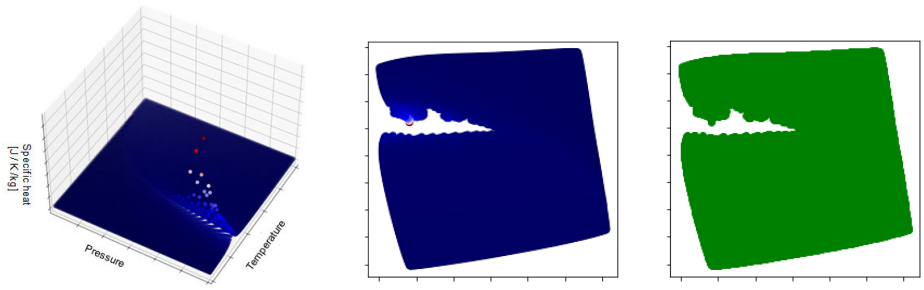
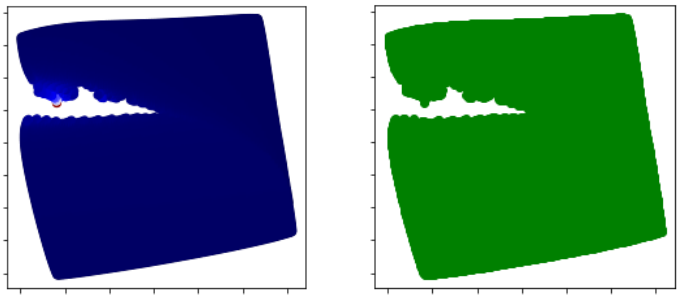


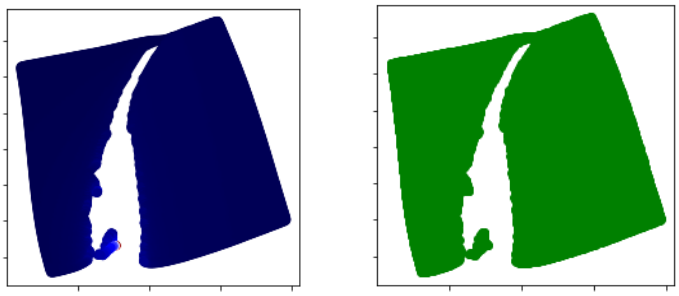
FIGURE 3.24: (a) Water specific-heat data; (b) t-SNE projection; (c) DBSCAN over t-SNE projection.

The automatic division method developed in Section 3.2.2.2 is also evaluated using the water specific-heat dataset. The process is illustrated in Figure 3.24. The original 3D data (Figure 3.24 (a)) is projected to two dimensions using t-SNE (Figure 3.24 (b)). The algorithm successfully finds the manifold peak and aims at splitting the data following the highest specific-heat values.

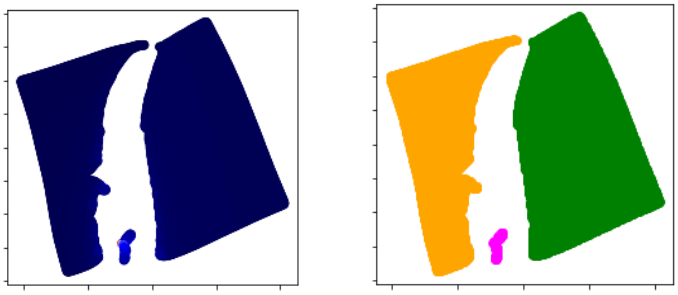
However, as pressure decreases, the data becomes more uniform. Therefore, the projected data is not completely divided and DBSCAN fails at identifying separated clusters (Figure 3.24 (c)).



(a) Iteration 1: t-SNE (b) Iteration 1: DBSCAN



(c) Iteration 2: t-SNE (d) Iteration 2: DBSCAN



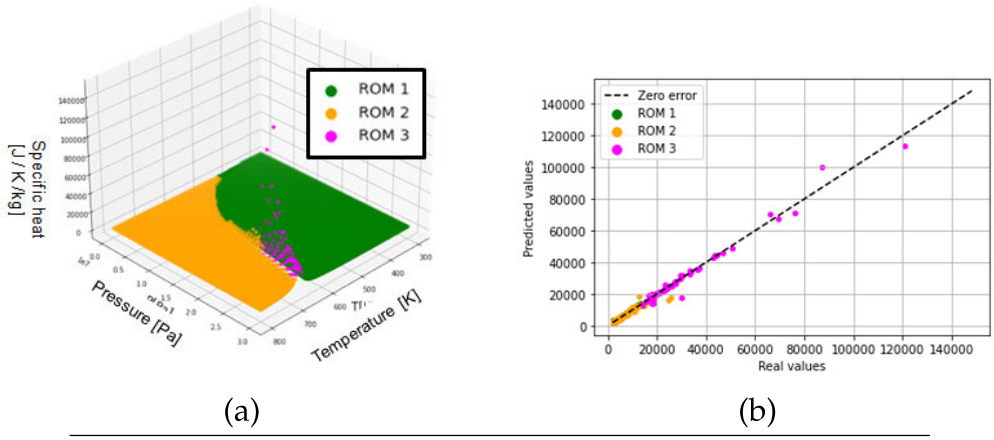
(e) Iteration 3: t-SNE (f) Iteration 3: DBSCAN

FIGURE 3.25: Automatic division iterative process.

The result of applying the automatic division to the water specific-heat dataset, consisting of a data projection using t-SNE and data grouping through DBSCAN is promising but incomplete.

The next step is to perform the automatic division once again over the projected data. It means, t-SNE is applied to the outcome of the first run of t-SNE, using the same parameters except for the output dimension, which is kept to two. The purpose is to emphasize the separation that was observed in the first projection until the DBSCAN clustering recognizes more than one group of data.

The iteration process is depicted in Figure 3.25. It is shown that the data is progressively separated with each projection until the DBSCAN can differentiate each group. Three iterations are required and three clusters are finally classified.



(a) (b)

FIGURE 3.26: (a) Automatic division in three clusters (b) ROM results.

The resulting clusters are identified in their original dimension in Figure 3.26 (a). Please note that while the automatic method on the water density case performed a similar separation than the knowledge-guided division, in this case, the highest values of specific heat are grouped. It means, the data surrounding the critical point is set as a separated cluster, but it does not correspond to the supercritical phase. However, except for cluster 3, the data corresponding to clusters 1 and 2 seem to follow the phase-based separation in two groups that was tested at the beginning of this section (see Figure 3.21 (b)).

The three automatically-divided datasets are used to train their respective ROMs, and the results of the real and predicted data are depicted in Figure 3.26 (b). Compared to the results of the knowledge-guided separations, where the highest values of the specific heat were severely underpredicted, this approach ensures better modelling of the manifold peak and the accuracy prediction reaches 99.5%.

3.2.4 Conclusions

In this section, a methodology to perform an automatic separation of manifolds with complexities is developed and tested. This procedure enhances the efficiency of tensor decomposition as a reduced order model when the

divided groups are trained individually vs training a single model using the original dataset. This approach outperforms supervised knowledge-based clustering. In addition, it can be applied to any n-dimensional space.

As the main disadvantage, it is required that the number of samples is significant since in a manifold of poor density a division could be mistakenly assigned to an empty area instead of manifold complexity.

3.3 Novel sequential DoE for ROM building

Reduced-order modelling aims at approximating the response of complex physical systems at minimum computational cost, avoiding running expensive high-fidelity simulations. Nevertheless, they usually require a comprehensive sampling to yield significant accuracy. In this section, a novel sampling algorithm is proposed whose purpose is to achieve the best model performance using the minimum number of samples. The method is described and applied to two test cases. The results are compared to two classical techniques of designs of experiments.

3.3.1 Design of experiments in computer simulation

The data to train a ROM with can be obtained from experiments or calculated using analytical equations or computer simulations. Regardless of the source of the data, a correct selection of the samples is essential to the success of the resulting model.

For that reason, the first step to building a ROM is to perform a design of experiments (DoE), which must guarantee a good training dataset in terms of both quality and quantity. It implies that data amount is sufficient and that samples are placed in such distribution that the model can appropriately capture the response of the system in detail. In this study, the term "experiment" in the design of experiments refers to the execution of a computer simulation model or analytical model. Therefore, unlike physical experimentation, problems such as noise or bias do not need to be addressed.

The optimum size of the training dataset can not be determined beforehand and it depends on other aspects, like data distribution and ROM configuration parameters. Usually, the ROM's performance will benefit from increasing the amount of data until certain maximum accuracy for that given setting is reached and more data will no longer improve it.

It is also impossible to estimate the best sampling distribution on a completely new case. That is why most classical DoE relies on the most intuitive criteria, which is space-filling. Factorial designs (full or fractional) [31] and central composite designs provide structured sampling distributions (grid designs)[86]. However, the number of samples required to complete them increases exponentially with the number of dimensions [23]. Thus random designs, such as Latin Hypercube or Montecarlo samplings, have become increasingly common. Latin Hypercube and Stratified Montecarlo designs are

stratified random samplings, that ensures more even distribution and avoid cluster generation compared to pure chaotic selections [38].

These sampling techniques are denoted as static, because they create the sampling dataset at once before any function evaluation is performed. Then each sample is calculated and from the set of input values and their response a surrogate model is built, as depicted in Figure 3.27 (a). Thus, no information about the output of the system contributes to determining the sample selection. Most static DoE's aim at uniformly covering the input space, which is known as space exploration. However, some methods are based on statistical criteria, usually for Gaussian process models, which allows a preliminary estimation of the model parameters. Some examples are maximum entropy designs or Mean Squared Prediction Error Designs [104, 38].

In the previous sections of the present chapter, it is assumed that the data to feed the ROM is sufficient and easily available. However, when the input data is generated through computer simulations, the amount of cases to run is generally limited by the available resources. High-fidelity simulations that aim at numerically replicating physical systems, usually involving several phenomena, are computationally expensive. Despite recent advances in digital computing, these simulations might take days to complete, depending on the complexity of the system. In this context, it is necessary to select a DoE that provides the maximum information using the minimum number of samples.

This goal is only achievable using a dynamic DoE that includes the response of the system and evolves accordingly, as opposed to the static DoE's described above. This type of DoE begins with an initial set of samples that are evaluated and used to train a surrogate model. If the model results do not meet the stop criteria, the training dataset is updated with a new sample or samples and the process is repeated (see Figure 3.27 (b)). This type of dynamic sample selection is known as adaptive DoE, sequential DoE or active learning.

The new sample is determined based on the information generated in the previous iterations. The increasing collection of function evaluations allows placing sample points in complex regions, such as non-linear areas, steep slopes or discontinuities, which is known as exploitation.

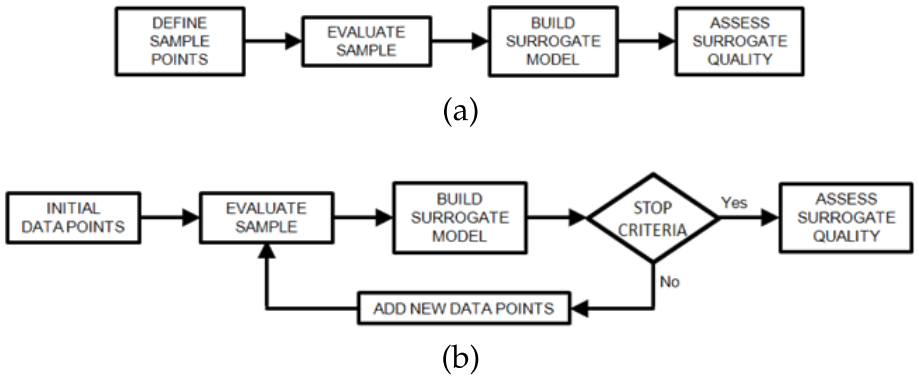


FIGURE 3.27: (a Static DoE (b) Sequential DoE.

If the initial sampling dataset contains a large number of homogeneously-distributed samples, the sequential DoE might rely only on exploitation criteria [32] or exploration criteria, such Voroni-based or Delaunay-based sequential design [26]. However, most adaptive designs aim at balancing exploration and exploitation simultaneously.

Different strategies have been developed in the last few years to select the next design point in sequential DoE. Crombecq et al. [25, 24] combined Voroni tessellations and local linear approximations (LOLA) as exploration and exploitation criteria respectively to create the LOLA-Voroni design of experiments. Some examples of applications of the LOLA-Voroni algorithm are found in the literature [30, 128].

Singh et al. [113] roposed a modification for adapting the ratio of exploration and exploitation components. It consists in adding a tuning parameter that weighs both functions according to three different balancing schemes.

Garud et al. [39] presented a smart sampling algorithm for unidimensional functions based on optimizing a crowding distance metric with a departure function. The first evaluates sample distribution while the second measures the quality of the surrogate model. The same author improved this technique by implementing Delaunay triangulation as an exploration method, allowing to choose the most promising subregion based on the local quality of the model [40].

Liu et al. [71] focused on developing an adaptive sampling for kriging by equilibrating an exploration term, based on the prediction variance, and an exploitation term, based on the maximum prediction error of the model, through a balance factor. Kleijnen et al. [63] also use the variance of the

predicted output of the kriging model calculated for several candidates, previously selected by a space-filling DoE.

The main drawback of these variance-based DoE's is that they are model-dependent. Thus, some works propose a similar approximation but replacing the Kriging model variance estimation with the prediction variance calculated using the cross-validation error. For instance, Eason et al. [32] presented this approach along with an artificial neural network as surrogate model, while Wang et al. [133] combined LOO (leave-one-out) cross-validation with Voroni diagrams as exploration criteria to create the Voroni-CV sampling algorithm.

These are representative examples that summarize the main approximations to exploration and exploitation criteria definition; however, extensive references can be found in literature reviews [72, 23, 38].

In this work, a new approach for sequential sampling is presented. The exploitation criteria of the sequential DoE presented in this work is based on a function that estimates the gradient around the sampled regions. A distance function penalizes the gradient-based function to avoid local oversampling, accounting for space exploration.

3.3.2 Methodology of novel sequential DoE

The sequential DoE presented in this work starts with an initial sample set and increases progressively by adding a new sample in each iteration. The new sample is selected by balancing the space-filling and the variations of the response surface. Thus it is ensured that there are no empty areas in the sampling domain and that enough information is provided where rapid changes occur.

The workflow of the algorithm is represented in Figure 3.28. Next, each step of the process is described in depth.

1. To begin with, the specifications of the dataset must be defined. In particular, the number of dimensions of the input space and the limits of each dimensión. It must be reminded that the output is univariate and that the accuracy of the ROM prediction is only guaranteed within the domain bounds.

2. An initial sampling set must be generated. It must include the input space boundaries, which corresponds to a 2-level full factorial, and at

least one internal sample. Therefore, the minimum size of this sampling set must be $2^n + 1$, being n the number of input dimensions. However, an initial dataset of too few samples slows the search for the optimal design points during the first iterations, since even the sequential DoE has too little information to work with. A full factorial of 3 or 4 levels is recommended to obtain a preliminary space exploration. On the other hand, if the input space dimensionality is high, it is more feasible to combine a 2-level full factorial with a Latin Hypercube sampling or any other design of experiments. In addition, an irregular sampling distribution is advantageous for the sequential DoE performance during the first iterations.

3. Once that the initial sampling set is defined, for each sample x_i the corresponding real response of the output variable $f(x_i)$ is calculated. The final training dataset Ω_u is constituted by the initial set of samples and their output.

4. A distance function F_d over the dataset Ω_u is computed to account for the space exploration criteria. The distance among existing samples is calculated based on the euclidean norm. Therefore, the value of this function is maximum in the intermediate points among samples and zero in the samples themselves. Thus, the DoE avoids placing a new sample over an existing one.

5. A surrogate model S_0 is calculated using the complete dataset Ω_u. Regarding the definition of the TWINKLE parameters, they can be modified for each case as long as they are consistent through all the DoE iterations. The discretization net is designed through the adapted discretization method (see Appendix A).

6. The algorithm loops over all the internal sampling points of the training dataset Ω_u. For each inner sample P_i, a new dataset Ω_i is created, where sample P_i is removed from Ω_u.

7. A new surrogate model S_i is built using the reduced training dataset Ω_i. The ROM configuration is the same as in S_0.

8. The objective function Z_i is defined as the product of the distance function F_d and the difference function $|S_0 - S_i|$ among the two surrogate models. This difference function evaluates the effect of placing a new sample point in the area surrounding P_i, representing sampling space

exploitation. By weighing it with the distance function, it is balanced with the exploration criteria.

9. The objective function Z_i is maximized to find the sample point x_i that best equilibrates exploration and exploitation as defined by this procedure. Thus, if the difference function is greater than the distance function, it would correspond to a point close to P_i but where the gradient of the response surface is large, so it is advantageous to place a sample to better model an area with rapid changes. Where the distance function is larger than the difference function, it indicates that there is an extensive space fraction that is empty and it is convenient to fill it.

10. A sample candidate x_i is determined under the premise of removing P_i. The process is repeated consecutively for all inner sample points, from steps 6 to 10. For each iteration, a sample candidate x_i is calculated. Finally, a list of sample candidates and their scores in the objective function $Z_i(x_i)$ is obtained.

11. Through all sample candidates x_i, the selected next sample point x_{max} is the sample candidate x_i whose score value in the objective function $Z_i(x_i))$ is maximum. The output of this design point x_{max} is calculated and both the sample and its evaluation are added to the ROM training dataset.

12. Finally, the stop criteria are checked. The user must specify these stop criteria beforehand and adapt them according to the case circumstances. If the computational resources are limited, the DoE can stop generating new points if a fixed number of new samples is reached. If the model quality is essential, the DoE can stop if the ROM computed using the current dataset (including the new sample point) accomplish sufficient accuracy. If the stop criteria are not met, the last dataset Ω_u is updated adding the sample x_{max} and its output. The procedure is then started over from step 4 with the new data set Ω_u.

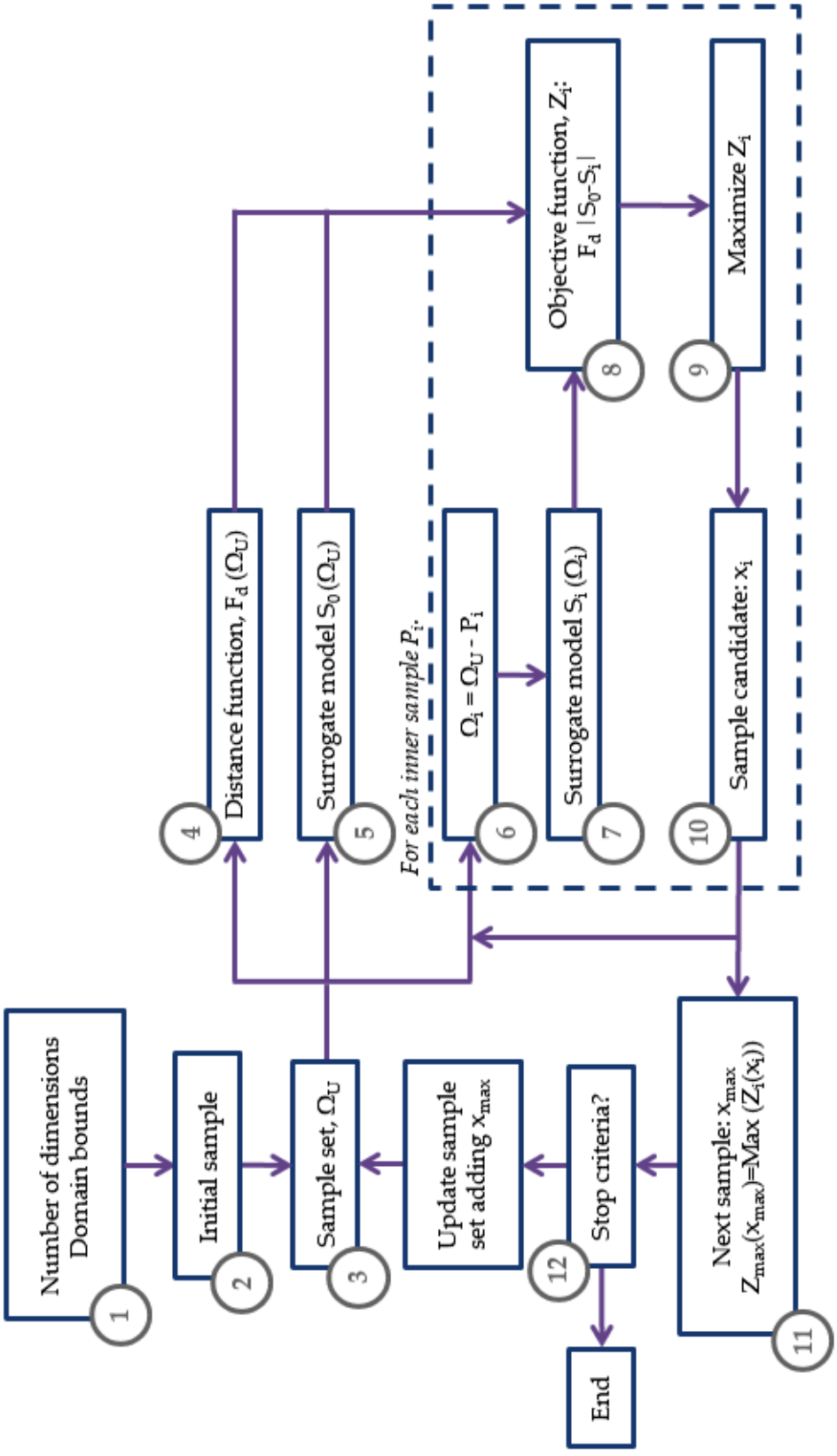


FIGURE 3.28: Flow chart of sequential DoE algorithm.

3.3.3 Test cases

The efficiency of the presented sequential design of experiments is evaluated using two analytical test functions: Rosenbrock and Easom functions. The performance of the obtained sampling dataset are compared to other two sampling techniques: full factorial and Latin hypercube sampling.

3.3.3.1 Rosenbrock function

The Rosenbrock function is a popular test problem for optimization algorithms. Although the function is defined for n dimensions, in this work the two-dimensional form is used:

$$f(x_1, x_2) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2 \tag{3.5}$$

The domain range is set to [-5,5] for x_1 and x_2. The Rosenbrock function is represented in Figure 3.29.

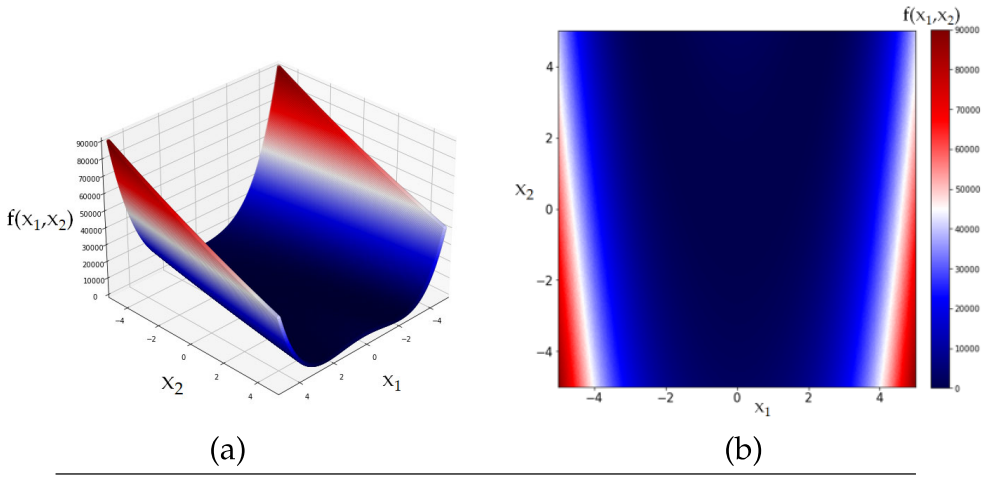


FIGURE 3.29: (a) Rosenbrock function (b) Rosenbrock function contours.

The initial sampling set is generated using a Full Factorial with four levels. The three-dimensional representation, the two-dimensional function contours and the distance function for this initial sampling are depicted in Figure 3.30.

The algorithm starts to calculate new sample points according to the described methodology. In Figure 3.31 the results of 20 samples dataset is shown, it means, after adding four samples using the sequential procedure.

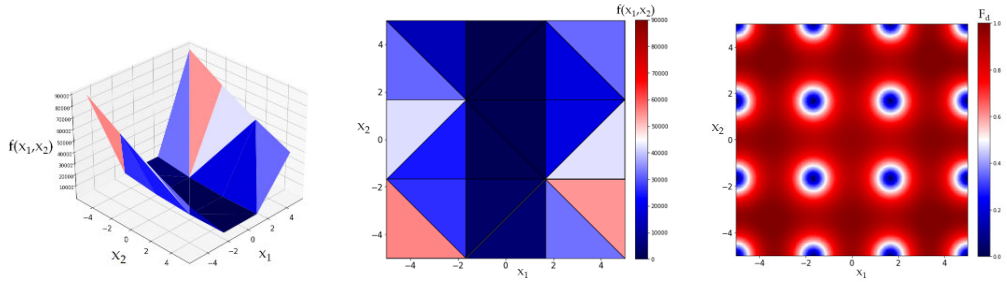


FIGURE 3.30: Rosenbrock function: initial sampling (a) 3d (b) Contours (c) Distance.

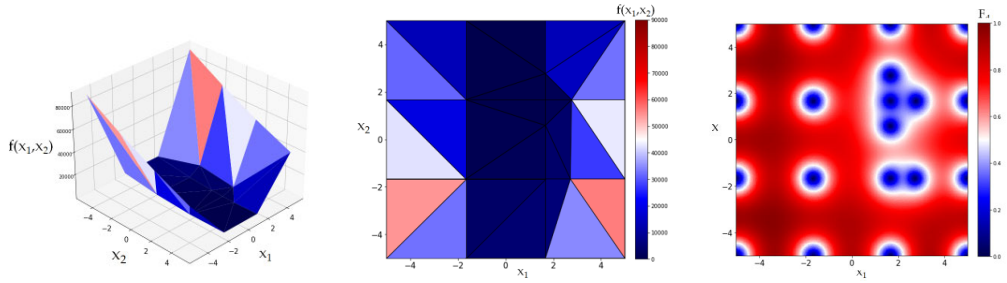


FIGURE 3.31: Rosenbrock function: 20 samples (a) 3d (b) Contours (c) Distance.

After the addition of 9 points, it can be observed in Figure 3.32 has located the large gradients at the lowes and highest values of x_1.

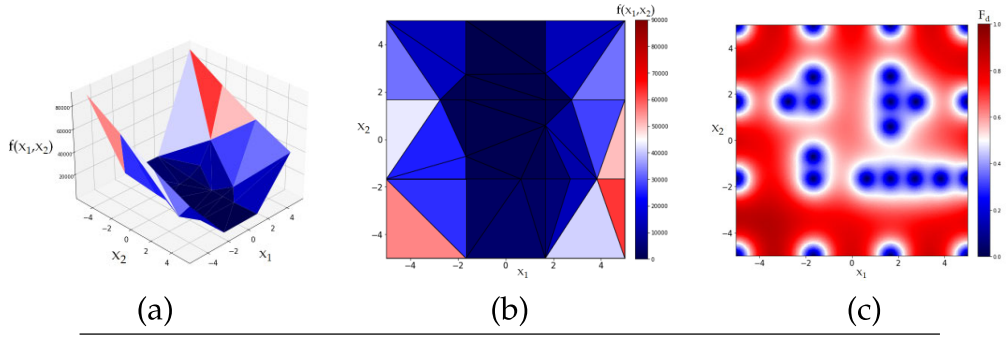
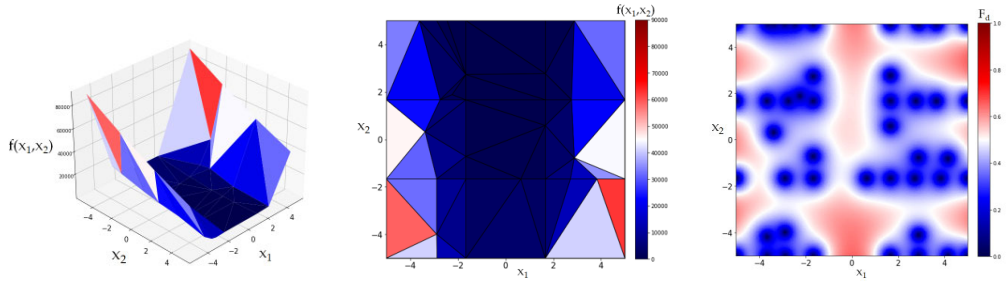


FIGURE 3.32: Rosenbrock function: 25 samples (a) 3d (b) Contours (c) Distance.

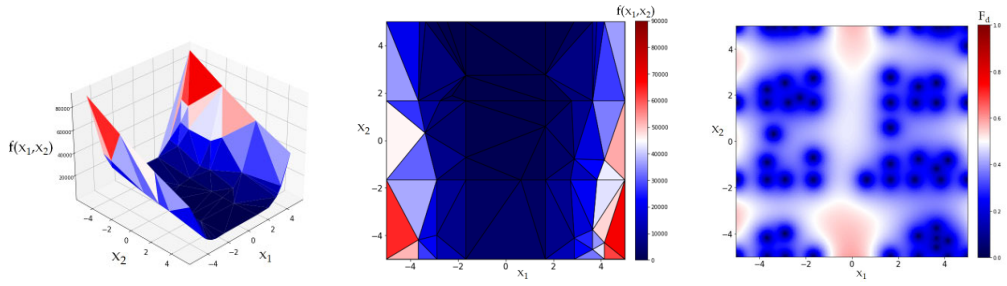
This tendency becames more clear in Figure 3.33, where 19 samples have been added through the sequential sampling to obtain a total of 35 points. All of them, except one, have been placed from -4 to -2 and from 2 to 4 in x_1.

With 50 samples, the shape of the function highly ressembles the original representation, see Figure 3.34.



(a) (b) (c)

FIGURE 3.33: Rosenbrock function: 35 samples (a) 3d (b) Contours (c) Distance.



(a) (b) (c)

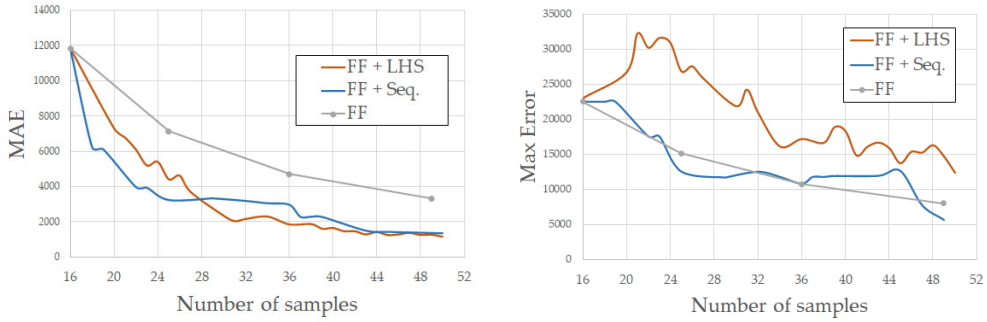
FIGURE 3.34: Rosenbrock function: 50 samples (a) 3d (b) Contours (c) Distance.

This final dataset of 50 samples is selected to be compared to other techniques of design of experiments: full factorial and Latin hypercube sampling. The starting dataset for the three methods is a full factorial of four levels. For the strictly full factorial procedure (FF), sampling datasets are generated using 5, 6 and 7 levels.

The Latin hypercube sampling is used to internally fill a domain that in this case is initiated with a full factorial of four levels (FF+LHS). Because it is a random design, it would not be appropriate to run the sampling once. Hence, this design of experiments is repeated 10 times and the accuracy results are averaged.

The obtained sampling datasets are evaluated using a test set that consists of a full factorial of 250 levels. The mean absolute error (MAE) and the maximum error are computed for models of increasing number of samples. The results are shown in Figure 3.35.

It can be observed that the MAE decreases faster with the number of samples for the smaller datasets. From 28 sampling points, the sequential and the LHS methods report similar absolute errors. However, the maximum errors of the



(a) (b)

FIGURE 3.35: (a) Mean absolute error (b) Maximum error.

LHS are significantly higher. The full factorial design shows the opposite tendency: the maximum errors are similar to the sequential technique, but the absolute errors are higher. Overall, the sequential algorithm shows the best performance.

It must be noted that 10 LHS were calculated and only the average values are reported. That means that there are more and less accurate LHS and to provide an estimation of that dispersion, the FF+LHS is represented with the standard deviation of the MAE in Figure 3.36.

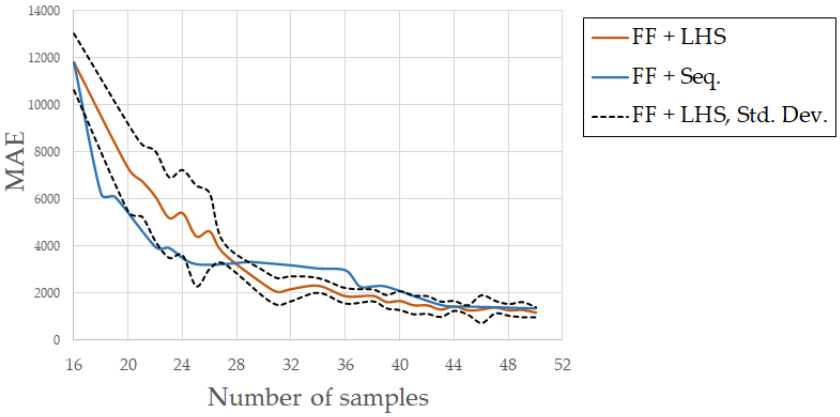


FIGURE 3.36: MAE of sequential sampling and LHS with standard deviation.

It is shown that for the smaller datasets the LHS could be as precise as the sequential sampling, but it could also present a noticeable worse performance, depending on the obtained LHS. Thus, a deterministic approach such as the sequential method is advantageous to ensure the robustness of the sampling.

3.3.3.2 Easom function

The Rosenbrock function presented moderate variations in the response surface across the specified domain. However, when exploring a completely new case, it could be possible that the region with the largest changes is located in a concrete subdomain. In that situation, space-filling DoEs are especially unfavourable. To represent this situation using an analytical function, the sequential DoE is tested for the Easom function, which is formulated as follows:

$$f(x_1, x_2) = -cos(x_1)cos(x_2)exp(-(x_1 - \pi)^2 + (x_2 - \pi)^2) \tag{3.6}$$

The domain boundaries are set to [-10,10] for x_1 and x_2. The test function for this case is represented in Figure 3.37.

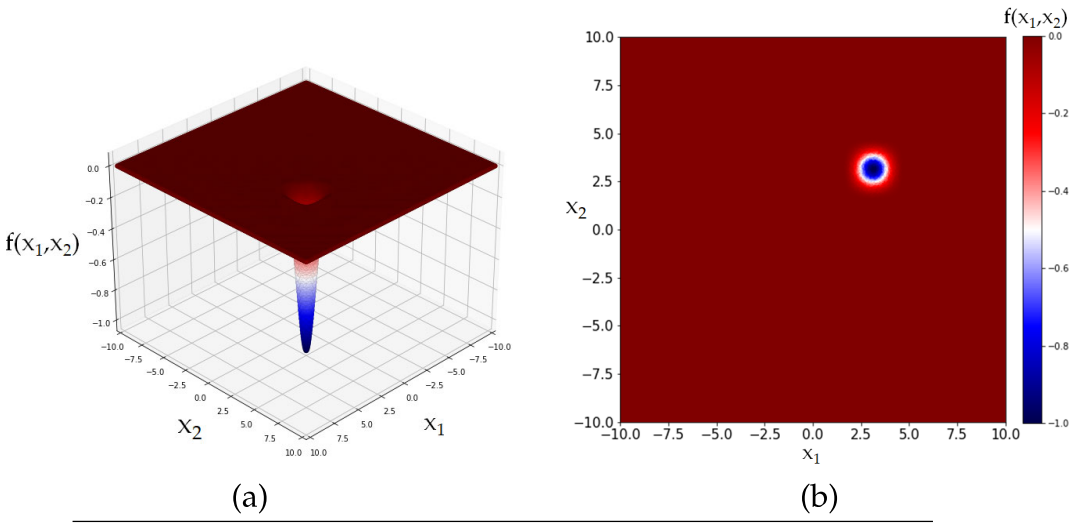


FIGURE 3.37: (a) Easom function (b) Easom function contours.

The initial sampling is design as a full factorial of 5 levels, as depicted in Figure 3.38. It must be observed that a ROM trained over this dataset would ignore the peak of the function and the prediction accuracy would be very poor.

The evolution of the sequential algorithm is illustrated next. After adding 10 sampling points through the sequential algorithm the peak has been already noticed, see Figure 3.39.

The modelling of the peak improves with increasing sampling points, as depicted in Figure 3.40 and 3.41.

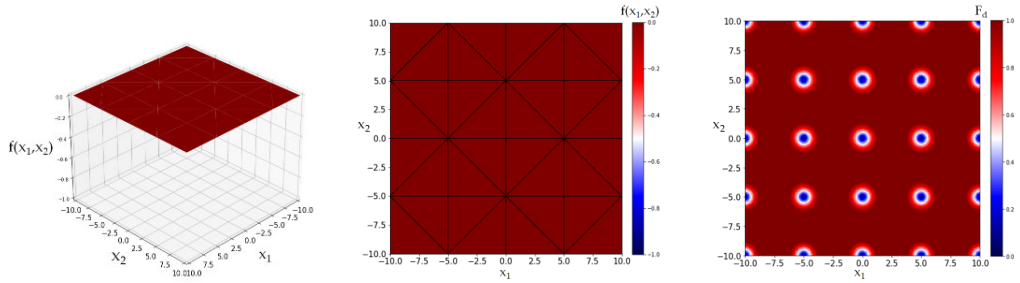


FIGURE 3.38: Easom function: initial sampling (a) 3d (b) Contours (c) Distance.

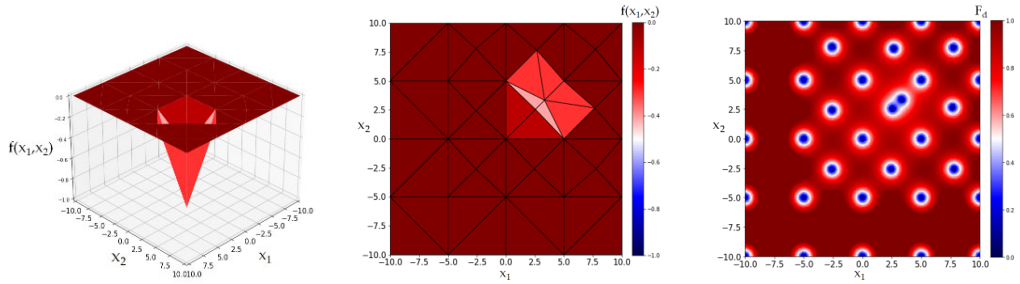


FIGURE 3.39: Easom function: 35 samples (a) 3d (b) Contours (c) Distance.

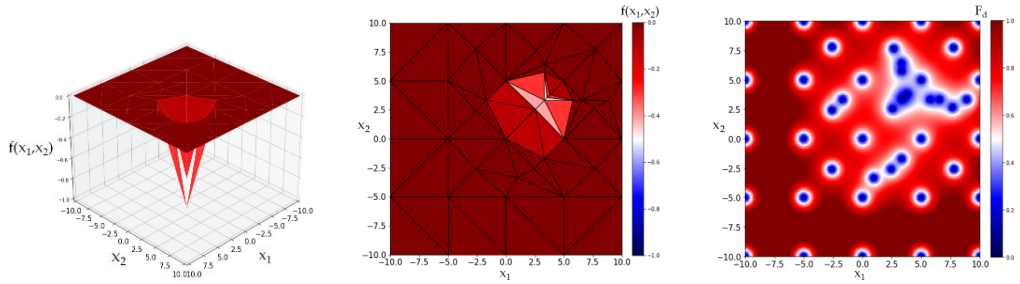


FIGURE 3.40: Easom function: 45 samples (a) 3d (b) Contours (c) Distance.

The resulting sampling dataset is compared to FF and FF+LHS samplings as described in the previous test function. In this case, the initial dataset is a FF of 5 levels. The results of mean absolute error (MAE) and maximum error are reported in Figure 3.42.

The observations are similar to the previous case: regarding MAE, the sequential algorithm provides a more efficient sampling using few samples compared to the LHS. The reason why FF is with at 36 and 49 samples is because the uniform distribution coincidentally places a sampling point close

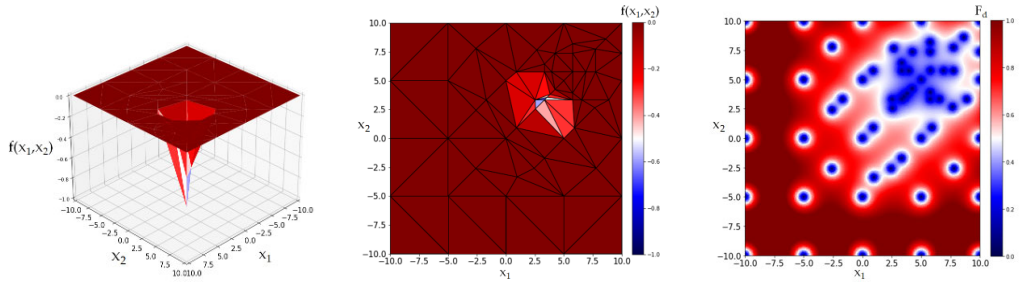


FIGURE 3.41: Easom function: 60 samples (a) 3d (b) Contours (c) Distance.

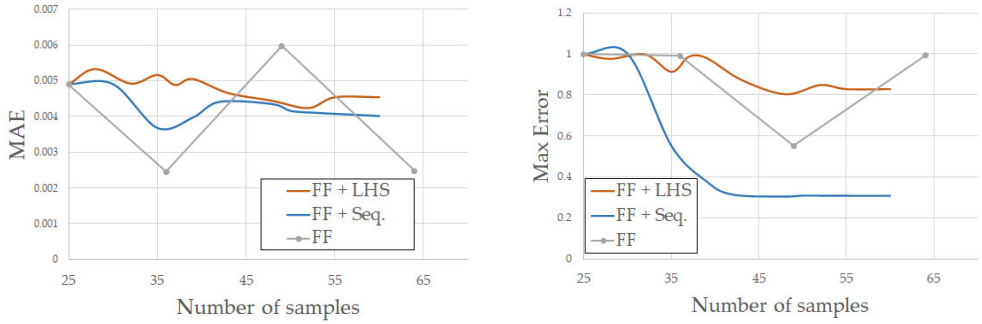


FIGURE 3.42: (a) Mean absolute error (b) Maximum error.

to the function minimum.

The surpassing performance of the sequential algorithm compared to space-filling DoE is clear in Figure 3.42 (b). The maximum error decreases significantly after approximately five samples are added sequentially.

The robustness of LHS is also checked in this test function. The results of the standard deviation of the MAE show the opposite tendency that in the Rosenbrock function: the dispersion increases with larger training datasets. This is probably a consequence of the randomization in LHS: with 55 sampling points some LHS will have capture the peaks but others will miss it, thus reporting high errors.

3.3.4 Conclusions

A novel sequential sampling strategy is developed to maximize the efficiency of the design of experiments step in surrogate modelling. As opposed to traditional space-filling criteria, this method includes the output of the system

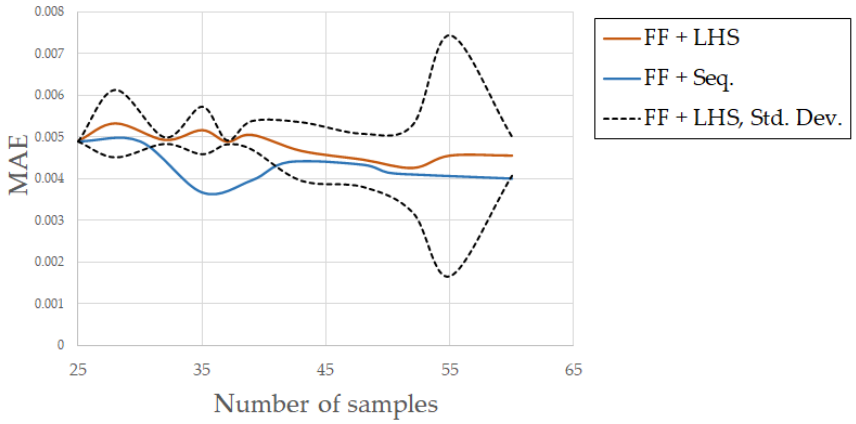


FIGURE 3.43: MAE of sequential sampling and LHS with standard deviation.

to identify the regions of the input domain that require a deeper examination. To avoid local oversampling, the exploitation criteria is balanced with a distance-based function.

A significant advantage of this procedure is that both the sequential sampling and the reduced order modelling run using the TWINKLE library. Thus the need for additional libraries or other dependencies is eliminated and it can be easily coupled to any computer simulation software.

The resulting training datasets generated using the sequential sampling are evaluated and compared to two popular designs of experiments: full-factorial and Latin hypercube sampling. The selected test functions are the Rosenbrock and the Easom function. In both cases, the surrogate models constructed from sequentially-calculated samples outperforms the full-factorial and LHS-based models, especially with small training datasets.

Chapter 4

Conclusions

This Thesis has presented several strategies to promote the applicability of numerical simulations in industrial environments from two different perspectives.

In the first part (Chapter 2), the development of predictive models for batch manufacturing processes is addressed using different approaches. The main features of each model are discussed using a rubber compounding process as a case study.

- A simplified theoretical model of the mixer is defined using a system of ordinary differential equations. A CFD simulation is performed for validation.
- A set of data-based models are proposed combining feature-extracted and feature-selected dimensionality reduction techniques and multivariate regression methods.
- Two additional cases of material processing are assessed to illustrate the differences between modelling with on-line, in-line and off-line data sources.

The most relevant conclusions are listed next:

- The accuracy of the theoretical approach is severely limited by two aspects: the formulated hypothesis for simplicity and incomplete information of the process. However, the model provides useful insight into the mixing process and the causal relationship among the process variables. This information can be leveraged for feature-selected data-driven models.
- Feature-extracted dimensionality reduction results in more accurate models. Nevertheless, models based on feature selection have physical

interpretability and are less data-intensive, which leads to more robust industrial implementations.

- The described methodologies can be used for the development of predictive models in other industrial processes, but each case must be carefully studied to efficiently adapt them to the particularities of each case. Specifically, the modelling approach highly depends on the availability of the process data and the process location.

The second part (Chapter 3) is focused on enhancing the coupling of surrogate modelling and numerical simulations through the development of different tools.

- A ROM implementation in a commercial CFD code is created for the determination of thermodynamic properties of several pure fluids and mixtures. This approach provides a robust and accurate coupling of equations of state into CFD simulations. The ROM interface programmed in this work is universal and can be easily used in different problems.

- A strategy to improve the accuracy of ROMs of complex manifolds is provided. It consists of dividing the domain and fitting a separated model to each region. These complexities represent particular dynamics of a physical system. Hence, a knowledge-based separation is suitable to achieve a successful domain split. In addition, a data-based division is proposed that outperforms the physical-guided separation. Both approaches are tested to model the density and specific heat of water.

- A novel sequential sampling method is designed for the efficient generation of training datasets for surrogate modelling. This procedure achieves accurate ROM using fewer samples than the traditional space-filling design of experiments. The algorithm is tested using two analytical functions: Rosenbrock and Easom function. The results show that the sequential approach outperforms other popular techniques (full factorial and Latin hypercube sampling).

Appendix A

Adapted discretization

TWINKLE is based on the idea that a problem of N variables can be decomposed into a product of one-dimensional functions. In the mathematical implementation, these functions are piece-wise linear functions. These one-dimensional functions are discretized at several points, where each function is evaluated. The combination of every list of discretization points in each dimension represents the discretization net.

ThIS discretization can be generated in different ways. The default option is to set the same number of discretization points for every dimension, which results in a uniform discretization grid. The number of discretization points for each dimension can also be specified, which divides every one-dimensional function into sections of equal length. The concrete values of each variable can also be provided, which is used to create a non-uniform discretization net. Finally, the discretization points can be set equal to the sampling points.

The uniform discretization option works well in cases where the response of the system is smooth and widespread across the domain. However, if a subregion presents rapid changes, a uniform discretization will only be successful if a discretization point is placed in or near that section. However, if the discretization is coarse and the steep slopes are far from any discretization points, the ROM will fail in capturing the real behaviour of the system and the accuracy will decrease, even with a large training dataset.

A discretization net adapted to the topology of the problem could deal with local steep gradients. In addition, it could reduce the required number of discretization points, which would accelerate the ROM evaluation. The time saving is not significant in a single evaluation, but it could be noticeable if the ROM is implemented in a node of a computational mesh, in which case

to solve the problem it would have to run thousands of iterations over every node in the domain.

In this work, an algorithm to create an adapted discretization net is proposed. The aim is to place more discretization points in the sections with larger gradients in the one-dimensional functions. To ensure that the function is sufficiently sampled, the distance among the discretization points is also considered. The discretization points are defined sequentially: in each iteration, the next point is determined. An optimal discretization net usually requires a different number of discretization points in each dimension. Hence, the algorithm is prepared to calculate the next discretization point across all the variables.

The methodology is tested using the water heat capacity dataset as a function of temperature and pressure. The pressure range is set to 2.45E07 Pa to 2.55E07 Pa and the temperature range is 298 to 800 K. Under these specifications, the dataset presents a large peak at the critical temperature, as depicted in Figure A.1.

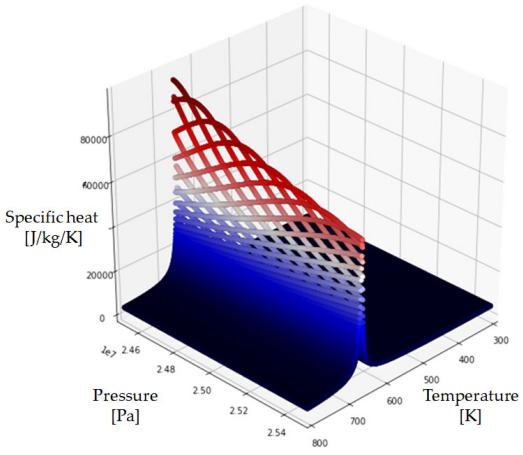


FIGURE A.1: Water heat capacity.

First, it is attempted to create a ROM from the described dataset using a uniform discretization. The training data is generated using a full factorial of 1000 levels. Three uniform discretizations are tested, using 50, 100 and 200 discretization points. It can be observed in Figure A.2 that, even refining the discretization net, the model fails at predicting the higher values of the heat capacity, placed at the peak. The maximum error for 200 discretization points is 17%.

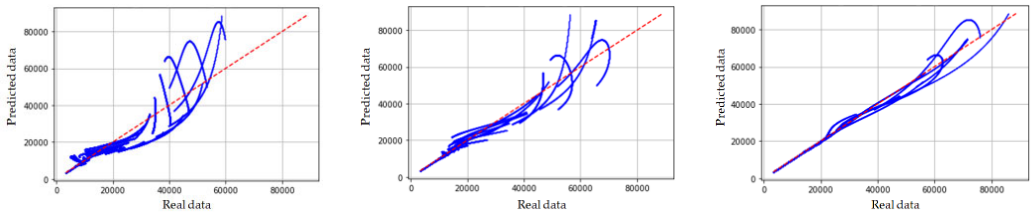


FIGURE A.2: ROM prediction using uniform discretization nets of (a) 50 points (b) 100 points and (c) 200 points.

Hence, the adapted discretization method is applied to this case to improve the accuracy of the model. First, an initial discretization net is defined. In this case, a uniform discretization net of eight points is set. Since the peak corresponds to a physical phenomenon, the critical region is identified. Thus, an extra discretization point is added, corresponding to the critical temperature (647 K). The discretization net over the domain is shown in Figure A.3.

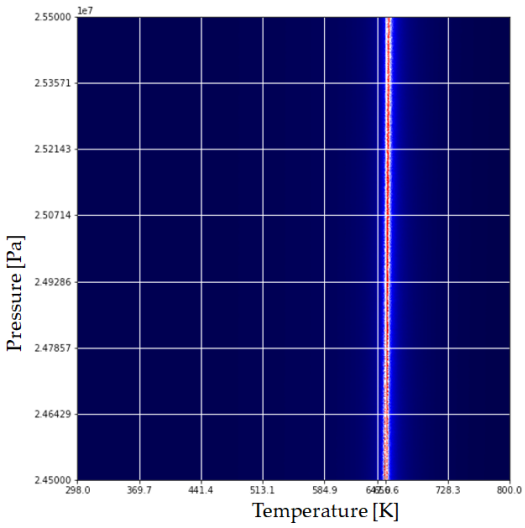


FIGURE A.3: Initial discretization net.

First, the dataset is decomposed into one-dimensional functions (f1d) through the TWINKLE library using the initial discretization previously provided. Hence, the temperature function consists of 9 points and 8 for the pressure. A distance function based on the euclidean norm is calculated for each variable. The value of this function is maximum at the intermediate distance of the furthest points and zero at the existing points. The distance function for the initial decomposition of the temperature is illustrated in Figure A.4.

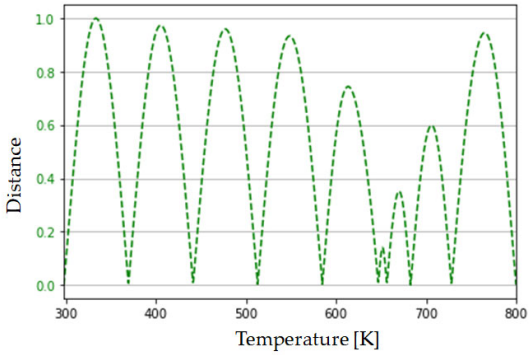


FIGURE A.4: Distance function for the temperature discretization.

Then, a difference function is computed for each internal discretization point of every variable. To achieve that, the discretization points of the individual one-dimensional functions are used to obtain a function S0. The function Si is calculated from all the discretization points except one internal point, which is removed. The difference function is the absolute difference between S0 and Si. In Figure A.5 an example of a difference function is found. In this case, the discretization point T= 656.57 K is eliminated from the temperature function to generate Si. It can be observed that the difference function is only quantitatively relevant between the previous and the next discretization point of the one that was removed.

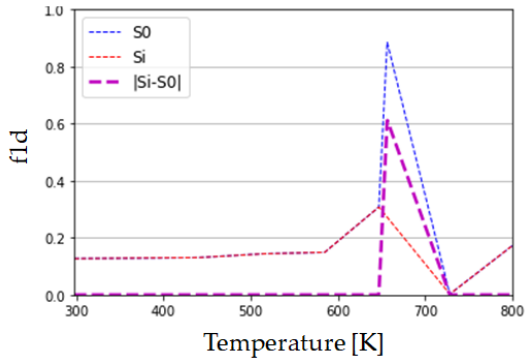


FIGURE A.5: Difference function for the temperature discretization.

Next, an objective function is defined as the product of the distance function and the difference function. The value of the one-dimensional function that achieves the maximum score of the objective function corresponds to the next proposed discretization point, see Figure A.6. It must be noted that this result is based on one extracted discretization point and one variable. Hence,

this procedure must be repeated for all internal discretization points of every variable. The proposed discretization points for each case is saved along with the corresponding value of the objective function. Finally, the discretization point that has the highest score in the objective function across all the internal points of every dimension is set as the final next discretization point and it is added to the list of discretization net. The process is repeated until a certain number of calculated discretization points are reached or the ROM's performance does not improve with finer discretization nets.

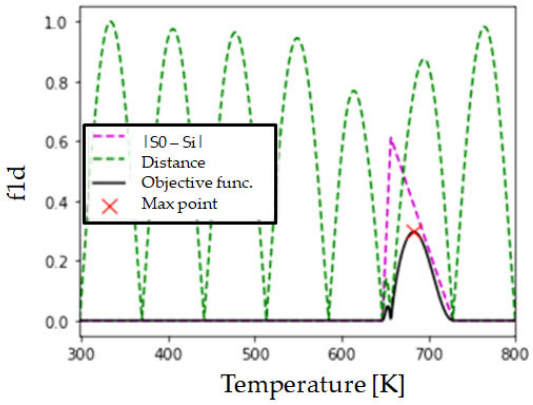


FIGURE A.6: Objective function and next proposed discretization point for temperature variable and extracted 656.57 K.

The improvement of the water heat-capacity model with increasing adapted discretization points is shown in Figure A.7. For every addition to the discretization net, the model is evaluated using the training dataset. The relative and maximum relative error decreases while R^2 increases greatly for the first 15 iterations and stabilizes after. The stop condition was that the maximum relative error should be below 5 %. This is achieved after 42 iterations. The final number of discretization points of the temperature and pressure are 46 and 13, respectively, as depicted in Figure A.8.

The results of the adapted discretization are compared to the best uniform discretization in Figure A.9. The maximum error has decreased from 17.1 % to 4.7 %. In addition, the number of discretization points has been reduced from 200 in each dimension to 46 and 13, representing a 75 % and 94 % of reduction.

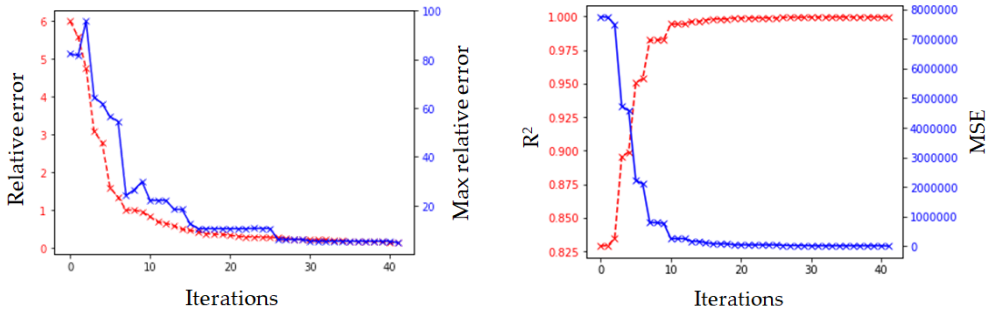


FIGURE A.7: (a) Relative and maximum relative error (b) R^2 and mean squared error (MSE).

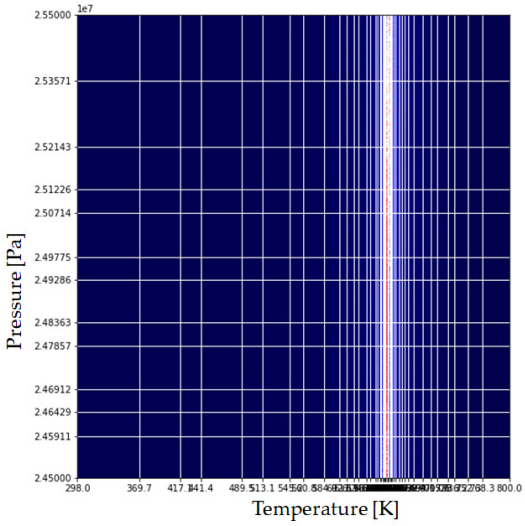


FIGURE A.8: Final discretization net.

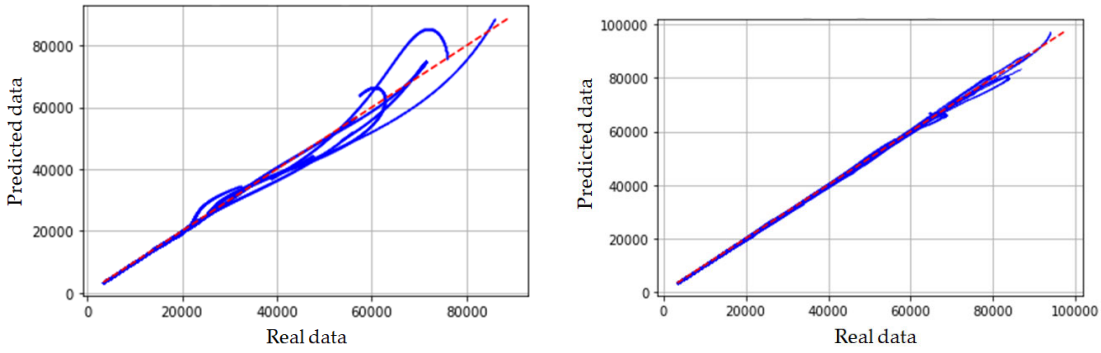


FIGURE A.9: ROM prediction using (a) uniform discretization net of 200 points; (b) adapted discretization points.

Bibliography

[1] Evrim Acar and Bülent Yener. "Unsupervised Multiway Data Analysis: A Literature Survey". In: *IEEE Transactions on Knowledge and Data Engineering* 21.1 (2009), pp. 6–20. DOI: 10.1109/TKDE.2008.112.

[2] Carmen Alfaro-Isac, Salvador Izquierdo-Estallo, and José Sierra-Pallares. "Reduced-order modelling of equations of state using tensor decomposition for robust, accurate and efficient property calculation in high-pressure fluid flow simulations". In: *The Journal of Supercritical Fluids* 165 (2020), p. 104938.

[3] Naseem Ali et al. "Cluster-based reduced-order descriptions of two phase flows". In: *Chemical Engineering Science* 222 (2020), p. 115660. ISSN: 0009-2509. DOI: https://doi.org/10.1016/j.ces.2020.115660. URL: https://www.sciencedirect.com/science/article/pii/S0009250920301925.

[4] David Amsallem, Matthew Zahr, and Charbel Farhat. "Nonlinear model order reduction based on local reduced-order bases". In: *International Journal for Numerical Methods in Engineering* 92 (Dec. 2012), pp. 891–916. DOI: 10.1002/nme.4371.

[5] Siam Aumi and Prashant Mhaskar. "Integrating data-based modeling and nonlinear control tools for batch process control". In: *AIChE journal* 58.7 (2012), pp. 2105–2119.

[6] Mehala Balamurali, Katherine L. Silversides, and Arman Melkumyan. "A comparison of t-SNE, SOM and SPADE for identifying material type domains in geological data". In: *Computers & Geosciences* 125 (2019), pp. 78–89. ISSN: 0098-3004. DOI: https://doi.org/10.1016/j.cageo.2019.01.011. URL: https://www.sciencedirect.com/science/article/pii/S0098300418306010.

[7] Daniel Banuti. "Crossing the Widom-line – Supercritical pseudo-boiling". In: *The Journal of Supercritical Fluids* 98 (Jan. 2015), pp. 12–16. DOI: 10.1016/j.supflu.2014.12.019.

[8] Ian H. Bell and Bradley K. Alpert. "Exceptionally reliable density-solving algorithms for multiparameter mixture models from Chebyshev expansion rootfinding". In: *Fluid Phase Equilibria* 476 (2018), pp. 89–102. ISSN: 0378-3812. DOI: https://doi.org/10.1016/j.fluid.2018.04.026. URL: https://www.sciencedirect.com/science/article/pii/S0378381218301730.

[9] Ian H. Bell et al. "Pure and Pseudo-pure Fluid Thermophysical Property Evaluation and the Open-Source Thermophysical Property Library CoolProp". In: *Industrial & Engineering Chemistry Research* 53 (2014), pp. 2498 –2508.

[10] Josette Bellan. "Theory, Modeling and analysis of turbulent supercritical mixing". In: *Combustion Science and Technology* 178.1-3 (2006), pp. 253–281. DOI: 10.1080/00102200500292241. eprint: https://doi.org/10.1080/00102200500292241. URL: https://doi.org/10.1080/00102200500292241.

[11] R. B. Bird, W. E. Stewart, and E. N. Lightfoot. *Transport phenomena*. John Wiley and Sons Inc., 1960. DOI: https://doi.org/10.1002/aic.690070245.

[12] Dominique Bonvin and Grégory François. *Control and optimization of batch chemical processes*. Tech. rep. Butterworth-Heinemann, 2017.

[13] Dominique Bonvin, Bala Srinivasan, and David Ruppen. *Dynamic optimization in the batch chemical industry*. Tech. rep. 2001.

[14] Giulio Borghesi and Josette Bellan. "A priori and a posteriori investigations for developing large eddy simulations of multi-species turbulent mixing under high-pressure conditions". In: *Physics of Fluids* 27.3 (2015), p. 035117. DOI: 10.1063/1.4916284. eprint: https://doi.org/10.1063/1.4916284. URL: https://doi.org/10.1063/1.4916284.

[15] Arianna Borrelli and Janina Wellmann. "Computer Simulations Then and Now: an Introduction and Historical Reassessment". In: *NTM* 27 (Dec. 2019), pp. 407–417. DOI: 10.1007/s00048-019-00227-6.

[16] Afef Ben Brahim and Mohamed Limam. "New prior knowledge based extensions for stable feature selection". In: *2014 6th International Conference of Soft Computing and Pattern Recognition (SoCPaR)*. IEEE. 2014, pp. 306–311.

[17] Arthur W Burks and Alice R Burks. "First general-purpose electronic computer". In: *IEEE Annals of the History of Computing* 3.04 (1981), pp. 310–389.

[18] F.A.R. Cardoso et al. "The use of polynomial models to determine thermodynamic properties of turbulent supercritical mixture in SAS process: A statistical analysis". In: *The Journal of Supercritical Fluids* 145 (2019), pp. 228–242. ISSN: 0896-8446. DOI: https://doi.org/10.1016/j.supflu.2018.12.007. URL: https://www.sciencedirect.com/science/article/pii/S0896844618306375.

[19] PJ Carreau, DCR De Kee, and RP Chhabra. *Rheology of Polymeric Systems, Principles and Applications. Hanser*. American Institute of Chemical Engineers: New York, NY, USA, 1997.

[20] Girish Chandrashekar and Ferat Sahin. "A survey on feature selection methods". In: *Computers & Electrical Engineering* 40.1 (2014), pp. 16–28.

[21] Gang Chen et al. "Support-vector-machine-based reduced-order model for limit cycle oscillation prediction of nonlinear aeroelastic system". In: *Mathematical problems in engineering* 2012 (2012).

[22] A. Cichocki et al. *Nonnegative Matrix and Tensor Factorizations: Applications to Exploratory Multi-way Data Analysis and Blind Source Separation*. Wiley, 2009. ISBN: 9780470747285. URL: https://books.google.es/books?id=KaxssMiWgswC.

[23] Karel Crombecq, Eric Laermans, and Tom Dhaene. "Efficient space-filling and non-collapsing sequential design strategies for simulation-based modeling". In: *European Journal of Operational Research* 214.3 (2011), pp. 683–696.

[24] Karel Crombecq et al. "A novel hybrid sequential design strategy for global surrogate modeling of computer experiments". In: *SIAM Journal on Scientific Computing* 33.4 (2011), pp. 1948–1974.

[25] Karel Crombecq et al. "A novel sequential design strategy for global surrogate modeling". In: *Proceedings of the 2009 winter simulation conference (WSC)*. IEEE. 2009, pp. 731–742.

[26] Karel Crombecq et al. "Space-filling sequential design strategies for adaptive surrogate modelling". In: *The first international conference on soft computing technology in civil, structural and environmental engineering*. Vol. 38. 2009.

[27] Wannes De Groote et al. "Neural Network Augmented Physics Models for Systems with Partially Unknown Dynamics: Application to Slider-Crank Mechanism". In: *IEEE/ASME Transactions on Mechatronics* (2021), pp. 1–1. DOI: 10.1109/TMECH.2021.3058536.

[28] I. De Marco et al. "Interactions of phase equilibria, jet fluid dynamics and mass transfer during supercritical antisolvent micronization:

The influence of solvents". In: *Chemical Engineering Journal* 203 (2012), pp. 71–80. ISSN: 1385-8947. DOI: https://doi.org/10.1016/j.cej.2012.06.129. URL: https://www.sciencedirect.com/science/article/pii/S1385894712008595.

[29] John M Dealy and Kurt F Wissbrun. *Melt rheology and its role in plastics processing: theory and applications*. Springer Science & Business Media, 2012.

[30] Dirk Deschrijver et al. "Adaptive sampling algorithm for macromodeling of parameterized S-Parameter responses". In: *IEEE Transactions on Microwave Theory and Techniques* 59.1 (2010), pp. 39–45.

[31] Benjamin Durakovic. "Design of experiments application, concepts, examples: State of the art". In: *Periodicals of Engineering and Natural Sciences (PEN)* 5.3 (2017).

[32] John Eason and Selen Cremaschi. "Adaptive sequential sampling for surrogate model generation with artificial neural networks". In: *Computers & Chemical Engineering* 68 (2014), pp. 220–232. ISSN: 0098-1354. DOI: https://doi.org/10.1016/j.compchemeng.2014.05.021. URL: https://www.sciencedirect.com/science/article/pii/S0098135414001719.

[33] Stein Ove Erikstad. "Merging physics, big data analytics and simulation for the next-generation digital twins". In: 2017, pp. 141–151.

[34] Ettore Fadiga et al. "CoolFOAM: The CoolProp wrapper for OpenFOAM". In: *Computer Physics Communications* 250 (2020), p. 107047. ISSN: 0010-4655. DOI: https://doi.org/10.1016/j.cpc.2019.107047. URL: https://www.sciencedirect.com/science/article/pii/S0010465519303777.

[35] Michael C. Fu and Shane G. Henderson. "History of seeking better solutions, AKA simulation optimization". In: *2017 Winter Simulation Conference (WSC)*. 2017, pp. 131–157. DOI: 10.1109/WSC.2017.8247787.

[36] María García-Camprubí et al. "Numerical Approach for the Assessment of Micro-Textured Walls Effects on Rubber Injection Moulding". In: *Polymers* 13.11 (2021), p. 1739.

[37] James Gareth et al. *An introduction to statistical learning: with applications in R*. Spinger, 2013.

[38] Sushant S Garud, Iftekhar A Karimi, and Markus Kraft. "Design of computer experiments: A review". In: *Computers & Chemical Engineering* 106 (2017), pp. 71–95.

[39] Sushant S. Garud, Iftekhar A Karimi, and Markus Kraft. "Smart sampling algorithm for surrogate model development". In: *Computers & Chemical Engineering* 96 (2017), pp. 103–114.

[40] Sushant S. Garud, Nivethitha Mariappan, and Iftekhar A. Karimi. "Surrogate-based black-box optimisation via domain exploration and smart placement". In: *Computers & Chemical Engineering* 130 (2019), p. 106567. ISSN: 0098-1354. DOI: https://doi.org/10.1016/j.compchemeng.2019.106567. URL: https://www.sciencedirect.com/science/article/pii/S0098135419306027.

[41] Yipeng Ge et al. "Deep residual learning applied to real-gas thermodynamics". In: *AIP Conference Proceedings* 2040.1 (2018), p. 150004. DOI: 10.1063/1.5079207. eprint: https://aip.scitation.org/doi/pdf/10.1063/1.5079207. URL: https://aip.scitation.org/doi/abs/10.1063/1.5079207.

[42] Edward Glaessgen and David Stargel. "The digital twin paradigm for future NASA and US Air Force vehicles". In: *53rd AIAA/ASME/ASCE/AHS/ASC structures, structural dynamics and materials conference 20th AIAA/ASME/AHS adaptive structures conference 14th AIAA*. 2012, p. 1818.

[43] JCB Gonzaga et al. "ANN-based soft-sensor for real-time process monitoring and control of an industrial polymerization process". In: *Computers & chemical engineering* 33.1 (2009), pp. 43–49.

[44] Michael Grieves and John Vickers. "Digital twin: Mitigating unpredictable, undesirable emergent behavior in complex systems". In: *Transdisciplinary perspectives on complex systems*. Springer, 2017, pp. 85–113.

[45] Stephen P Gurden et al. "A comparison of multiway regression and scaling methods". In: *Chemometrics and Intelligent Laboratory Systems* 59.1-2 (2001), pp. 121–136.

[46] Isabelle Guyon and André Elisseeff. "An introduction to variable and feature selection". In: *Journal of machine learning research* 3.Mar (2003), pp. 1157–1182.

[47] Jiequn Han, Linfeng Zhang, et al. "Integrating machine learning with physics-based modeling". In: *arXiv preprint arXiv:2006.02619* (2020).

[48] Yan-Lin He and Qun-Xiong Zhu. "A novel robust regression model based on functional link least square (FLLS) and its application to modeling complex chemical processes". In: *Chemical Engineering Science* 153 (2016), pp. 117–128.

[49] Kostas D Housiadas. "An exact analytical solution for viscoelastic fluids with pressure-dependent viscosity". In: *Journal of Non-Newtonian Fluid Mechanics* 223 (2015), pp. 147–156.

[50] Yi-Xin Hua, Ya-Zhou Wang, and Hua Meng. "A numerical study of supercritical forced convective heat transfer of n-heptane inside a horizontal miniature tube". In: *The Journal of Supercritical Fluids* 52.1 (2010), pp. 36–46. ISSN: 0896-8446. DOI: https://doi.org/10.1016/j.supflu.2009.12.003. URL: https://www.sciencedirect.com/science/article/pii/S0896844609003830.

[51] Hrvoje Jasak, Aleksandar Jemcov, Zeljko Tukovic, et al. "OpenFOAM: A C++ library for complex physics simulations". In: *International workshop on coupled methods in numerical dynamics*. Vol. 1000. IUC Dubrovnik Croatia. 2007, pp. 1–20.

[52] Jean-Noël Jaubert, Romain Privat, and Fabrice Mutelet. "Predicting the phase equilibria of synthetic petroleum fluids with the PPR78 approach". In: *AIChE Journal* 56.12 (2010), pp. 3225–3235. DOI: https://doi.org/10.1002/aic.12232. eprint: https://aiche.onlinelibrary.wiley.com/doi/pdf/10.1002/aic.12232. URL: https://aiche.onlinelibrary.wiley.com/doi/abs/10.1002/aic.12232.

[53] Ian T Jolliffe and Jorge Cadima. "Principal component analysis: a review and recent developments". In: *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* 374.2065 (2016), p. 20150202.

[54] Eurika Kaiser et al. "Cluster-based reduced-order modelling of a mixing layer". In: *Journal of Fluid Mechanics* 754 (2014), pp. 365–414.

[55] Eurika Kaiser et al. "Sparsity enabled cluster reduced-order models for control". In: *Journal of Computational Physics* 352 (2018), pp. 388–409. ISSN: 0021-9991. DOI: https://doi.org/10.1016/j.jcp.2017.09.057. URL: https://www.sciencedirect.com/science/article/pii/S0021999117307301.

[56] Michael G Kapteyn and Karen E Willcox. "From physics-based models to predictive digital twins via interpretable machine learning". In: *arXiv preprint arXiv:2004.11356* (2020).

[57] Anuj Karpatne et al. "Physics-guided neural networks (pgnn): An application in lake temperature modeling". In: *arXiv preprint arXiv:1710.11431* (2017).

[58] Anuj Karpatne et al. "Theory-guided data science: A new paradigm for scientific discovery from data". In: *IEEE Transactions on knowledge and data engineering* 29.10 (2017), pp. 2318–2331.

[59] Hiroshi Kato et al. "A data assimilation methodology for reconstructing turbulent flows around aircraft". In: *Journal of Computational Physics* 283 (2015), pp. 559–581. ISSN: 0021-9991. DOI: https://doi.org/10.1016/j.jcp.2014.12.013. URL: https://www.sciencedirect.com/science/article/pii/S0021999114008213.

[60] Nikhil Ketkar. "Introduction to keras". In: *Deep learning with Python*. Springer, 2017, pp. 97–111.

[61] TH Khang and ZM Ariff. "Vulcanization kinetics study of natural rubber compounds having different formulation variables". In: *Journal of Thermal Analysis and Calorimetry* 109.3 (2012), pp. 1545–1553.

[62] Jack P.C. Kleijnen. "Verification and validation of simulation models". In: *European Journal of Operational Research* 82.1 (1995), pp. 145–162. ISSN: 0377-2217. DOI: https://doi.org/10.1016/0377-2217(94)00016-6. URL: https://www.sciencedirect.com/science/article/pii/0377221794000166.

[63] Jack PC Kleijnen and WCM Van Beers. "Application-driven sequential designs for simulation experiments: Kriging metamodelling". In: *Journal of the operational research society* 55.8 (2004), pp. 876–883.

[64] Dmitry Kobak and Philipp Berens. "The art of using t-SNE for single-cell transcriptomics". In: *Nature communications* 10.1 (2019), pp. 1–14.

[65] Gülser Köksal, Inci Batmaz, and Murat Caner Testik. "A review of data mining applications for quality improvement in manufacturing industry". In: *Expert systems with Applications* 38.10 (2011), pp. 13448–13467.

[66] Werner Kritzinger et al. "Digital Twin in manufacturing: A categorical literature review and classification". In: *IFAC-PapersOnLine* 51.11 (2018), pp. 1016–1022.

[67] Vipin Kumar and Sonajharia Minz. "Feature selection: a literature review". In: *SmartCR* 4.3 (2014), pp. 211–229.

[68] Hao-Yeh Lee et al. "Grade transition using dynamic neural networks for an industrial high-pressure ethylene–vinyl acetate (EVA) copolymerization process". In: *Computers & Chemical Engineering* 33.8 (2009), pp. 1371–1378.

[69] E. Lemmon, M. Huber, and M.O. McLinden. *NIST Standard Reference Database 23: Reference Fluid Thermodynamic and Transport Properties (REFPROP), Version 9.0*. 2010. URL: https://pages.nist.gov/REFPROP-docs/ (visited on 02/10/2020).

[70] George C Linderman and Stefan Steinerberger. "Clustering with t-SNE, provably". In: *SIAM Journal on Mathematics of Data Science* 1.2 (2019), pp. 313–332.

[71] Haitao Liu, Jianfei Cai, and Yew-Soon Ong. "An adaptive sampling approach for Kriging metamodeling by maximizing expected prediction error". In: *Computers & Chemical Engineering* 106 (2017), pp. 171–182.

[72] Haitao Liu, Yew Ong, and Jianfei Cai. "A Survey of Adaptive Sampling for Global Metamodeling in Support of Simulation-based Complex Engineering Design". In: *Structural and Multidisciplinary Optimization* 57 (Jan. 2018). DOI: 10.1007/s00158-017-1739-8.

[73] Honghua Liu et al. "Using t-distributed Stochastic Neighbor Embedding (t-SNE) for cluster analysis and spatial zone delineation of groundwater geochemistry data". In: *Journal of Hydrology* 597 (2021), p. 126146. ISSN: 0022-1694. DOI: https://doi.org/10.1016/j.jhydrol.2021.126146. URL: https://www.sciencedirect.com/science/article/pii/S0022169421001931.

[74] Mengnan Liu et al. "Review of digital twin about concepts, technologies, and industrial applications". In: *Journal of Manufacturing Systems* 58 (2021), pp. 346–361.

[75] Yi Liu and Junghui Chen. "Integrated soft sensor using just-in-time support vector regression and probabilistic analysis for quality prediction of multi-grade processes". In: *Journal of Process control* 23.6 (2013), pp. 793–804.

[76] Yi Liu, Zhengjiang Zhang, and Junghui Chen. "Ensemble local kernel learning for online prediction of distributed product outputs in chemical processes". In: *Chemical Engineering Science* 137 (2015), pp. 140–151.

[77] Yi Liu et al. "Ensemble deep kernel learning with application to quality prediction in industrial polymerization processes". In: *Chemometrics and Intelligent Laboratory Systems* 174 (2018), pp. 15–21.

[78] Edwin Lughofer et al. "Self-adaptive evolving forecast models with incremental PLS space updating for on-line prediction of micro-fluidic chip quality". In: *Engineering Applications of Artificial Intelligence* 68 (2018), pp. 131–151.

[79] Laurens Van der Maaten and Geoffrey Hinton. "Visualizing data using t-SNE." In: *Journal of machine learning research* 9.11 (2008).

[80] Jordan Mandel, Raghunandan Avula, and Edward V Prochownik. "Sequential analysis of transcript expression patterns improves survival prediction in multiple cancers". In: *BMC cancer* 20.1 (2020), pp. 1–14.

[81] Robert Marks. "The Palgrave Encyclopedia of Strategic Management". In: Palgrave Macmillan, London, Jan. 2016. Chap. Monte Carlo, pp. 1–4. DOI: 10.1057/978-1-349-94848-2_709-1.

[82] H. Meng et al. "Transport and dynamics of liquid oxygen droplets in supercritical hydrogen streams". In: *Journal of Fluid Mechanics* 527 (2005), 115–139. DOI: 10.1017/S0022112004003106.

[83] Hua Meng and Vigor Yang. "A unified treatment of general fluid thermodynamics and its application to a preconditioning scheme". In: *Journal of Computational Physics* 189.1 (2003), pp. 277–304. ISSN: 0021-9991. DOI: https://doi.org/10.1016/S0021-9991(03)00211-0. URL: https://www.sciencedirect.com/science/article/pii/S0021999103002110.

[84] Nicholas Metropolis et al. "The beginning of the Monte Carlo method". In: *Los Alamos Science* 15.584 (1987), pp. 125–130.

[85] Richard S. Miller, Kenneth G. Harstad, and Josette Bellan. "Direct numerical simulations of supercritical fluid mixing layers applied to heptane–nitrogen". In: *Journal of Fluid Mechanics* 436 (2001), 1–39. DOI: 10.1017/S0022112001003895.

[86] D.C. Montgomery. *Design and Analysis of Experiments*. John Wiley & Sons, Incorporated, 2017. ISBN: 9781119113478. URL: https://books.google.es/books?id=Py7bDgAAQBAJ.

[87] Marvin K. Nakayama. "Output Analysis for Simulations". In: *Proceedings of the 38th Conference on Winter Simulation*. WSC '06. Monterey, California: Winter Simulation Conference, 2006, 36–46. ISBN: 1424405017.

[88] Richard Nance and Robert Sargent. "Perspectives on the Evolution of Simulation". In: *Electrical Engineering and Computer Science* 50 (Feb. 2002). DOI: 10.1287/opre.50.1.161.17790.

[89] Elisa Negri, Luca Fumagalli, and Marco Macchi. "A review of the roles of digital twin in CPS-based production systems". In: *Procedia Manufacturing* 11 (2017), pp. 939–948.

[90] Nora Okong'o and Josette Bellan. "Direct numerical simulation of a transitional supercritical binary mixing layer: Heptane and nitrogen". In: *Journal of Fluid Mechanics* 464 (Aug. 2002). DOI: 10.1017/S0022112002008480.

[91] M. Oschwald and A. Schik. "Supercritical nitrogen free jet investigated by spontaneous Raman scattering". In: *Experiments in Fluids* 27.6 (1999), pp. 497–506. ISSN: 1432-1114. DOI: 10.1007/s003480050374. URL: https://doi.org/10.1007/s003480050374.

[92] Eric J Parish and Karthik Duraisamy. "A paradigm for data-driven predictive modeling using field inversion and machine learning". In: *Journal of Computational Physics* 305 (2016), pp. 758–774.

[93] Suraj Pawar et al. "Physics guided machine learning using simplified theories". In: *Physics of Fluids* 33.1 (2021), p. 011701. DOI: 10.1063/5.0038929. eprint: https://doi.org/10.1063/5.0038929. URL: https://doi.org/10.1063/5.0038929.

[94] F. Pedregosa et al. "Scikit-learn: Machine Learning in Python". In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.

[95] Ding-Yu Peng and Donald B. Robinson. "A New Two-Constant Equation of State". In: *Industrial & Engineering Chemistry Fundamentals* 15.1 (1976), pp. 59–64. ISSN: 0196-4313. DOI: 10.1021/i160057a011. URL: https://doi.org/10.1021/i160057a011.

[96] Rakesh Kumar Phanden, Priavrat Sharma, and Anubhav Dubey. "A review on simulation in digital twin for aerospace, manufacturing and robotics". In: *Materials today: proceedings* 38 (2021), pp. 174–178.

[97] Pavel Pudil and Jana Novovičová. "Novel methods for feature subset selection with respect to problem knowledge". In: *Feature extraction, construction and selection*. Springer, 1998, pp. 101–116.

[98] Qinglin Qi et al. "Digital Twin Service towards Smart Manufacturing". In: *Procedia CIRP* 72 (2018). 51st CIRP Conference on Manufacturing Systems, pp. 237–242. ISSN: 2212-8271. DOI: https://doi.org/10.1016/j.procir.2018.03.103. URL: https://www.sciencedirect.com/science/article/pii/S2212827118302580.

[99] Ashwin Raghavan and Ahmed F. Ghoniem. "Simulation of supercritical water–hydrocarbon mixing in a cylindrical tee at intermediate Reynolds number: Impact of temperature difference between

streams". In: *The Journal of Supercritical Fluids* 95 (2014), pp. 325–338. ISSN: 0896-8446. DOI: https://doi.org/10.1016/j.supflu.2014.09.030. URL: https://www.sciencedirect.com/science/article/pii/S0896844614003192.

[100] K. Ren et al. "Thermal field prediction for laser scanning paths in laser aided additive manufacturing by physics-based machine learning". In: *Computer Methods in Applied Mechanics and Engineering* 362 (2020), p. 112734. ISSN: 0045-7825. DOI: https://doi.org/10.1016/j.cma.2019.112734. URL: https://www.sciencedirect.com/science/article/pii/S0045782519306243.

[101] Jun-Hyok Ri, Un-Il Ri, and Hyon-Sik Hong. "Multiscale analysis of elastic-viscoplastic composite using a cluster-based reduced-order model". In: *Composite Structures* 272 (2021), p. 114209. ISSN: 0263-8223. DOI: https://doi.org/10.1016/j.compstruct.2021.114209. URL: https://www.sciencedirect.com/science/article/pii/S0263822321006711.

[102] T.G. Ritto and F.A. Rochinha. "Digital twin, physics-based model, and machine learning applied to damage detection in structures". In: *Mechanical Systems and Signal Processing* 155 (2021), p. 107614. ISSN: 0888-3270. DOI: https://doi.org/10.1016/j.ymssp.2021.107614. URL: https://www.sciencedirect.com/science/article/pii/S0888327021000091.

[103] Eckhardt Roger. "Stan Ulam, John Von Neumann and the Monte Carlo Method". In: *Argonne, USA* (1987).

[104] Thomas J Santner et al. *The design and analysis of computer experiments*. Vol. 1. Springer, 2003.

[105] Robert G. Sargent. "A perspective on fifty-five years of the evolution of scientific respect for simulation". In: *2017 Winter Simulation Conference (WSC)*. 2017, pp. 3–15. DOI: 10.1109/WSC.2017.8247317.

[106] Guodong Shao et al. "Digital Twin for Smart Manufacturing: The Simulation Aspect". In: *2019 Winter Simulation Conference (WSC)*. 2019, pp. 2085–2098. DOI: 10.1109/WSC40007.2019.9004659.

[107] José Sierra-Pallares, Ashwin Raghavan, and Ahmed F. Ghoniem. "Computational study of organic solvent–CO2 mixing in convective supercritical environment under laminar conditions: Impact of enthalpy of mixing". In: *The Journal of Supercritical Fluids* 109 (2016), pp. 109–123. ISSN: 0896-8446. DOI: https://doi.org/10.1016/

j.supflu.2015.11.007. URL: https://www.sciencedirect.com/science/article/pii/S0896844615301807.

[108] José Sierra-Pallares, Pablo Santiago-Casado, and Francisco Castro. "Numerical modelling of supercritical submerged water jets in a subcritical co-flow". In: *The Journal of Supercritical Fluids* 65 (2012), pp. 45–53. ISSN: 0896-8446. DOI: https://doi.org/10.1016/j.supflu.2012.02.028. URL: https://www.sciencedirect.com/science/article/pii/S0896844612000770.

[109] José Sierra-Pallares et al. "A computational fluid dynamics study of supercritical antisolvent precipitation: Mixing effects on particle size". In: *AIChE Journal* 58.2 (2012), pp. 385–398. DOI: https://doi.org/10.1002/aic.12594. eprint: https://aiche.onlinelibrary.wiley.com/doi/pdf/10.1002/aic.12594. URL: https://aiche.onlinelibrary.wiley.com/doi/abs/10.1002/aic.12594.

[110] José Sierra-Pallares et al. "Numerical modelling of hydrothermal flames. Micromixing effects over turbulent reaction rates". In: *The Journal of Supercritical Fluids* 50.2 (2009), pp. 146–154. ISSN: 0896-8446. DOI: https://doi.org/10.1016/j.supflu.2009.05.001. URL: https://www.sciencedirect.com/science/article/pii/S0896844609001703.

[111] José Sierra-Pallares et al. "Quantification of mixing efficiency in turbulent supercritical water hydrothermal reactors". In: *Chemical Engineering Science* (Apr. 2011). DOI: 10.1016/j.ces.2010.12.045.

[112] José Sierra-Pallares et al. "Understanding bottom-up continuous hydrothermal synthesis of nanoparticles using empirical measurement and computational simulation". In: *Nano Research* 9.11 (2016), pp. 3377–3387. ISSN: 1998-0000. DOI: 10.1007/s12274-016-1215-6. URL: https://doi.org/10.1007/s12274-016-1215-6.

[113] Prashant Singh, Dirk Deschrijver, and Tom Dhaene. "A balanced sequential design strategy for global surrogate modeling". In: *2013 Winter Simulations Conference (WSC)*. IEEE. 2013, pp. 2172–2179.

[114] Roland Span. *Multiparameter equations of state : an accurate source of thermodynamic property data : with 151 figures and tables /*. Jan. 2000. ISBN: 978-3-642-08671-7. DOI: 10.1007/978-3-662-04092-8.

[115] Jan Östh et al. "Cluster-based reduced-order modelling of the flow in the wake of a high speed train". In: *Journal of Wind Engineering and Industrial Aerodynamics* 145 (2015), pp. 327–338. ISSN: 0167-6105. DOI: https://doi.org/10.1016/j.jweia.2015.06.003.

URL: https://www.sciencedirect.com/science/article/pii/S0167610515001415.

[116] Ali Takbiri-Borujeni and Mohsen Ayoobi. "Application of physics-based machine learning in combustion modeling". In: *11th US National Combustion Meeting*. 2019, p. 10.

[117] Yifei Tan et al. "Application of IoT-aided simulation to manufacturing systems in cyber-physical system". In: *Machines* 7.1 (2019), p. 2.

[118] Fei Tao et al. "Digital twin in industry: State-of-the-art". In: *IEEE Transactions on Industrial Informatics* 15.4 (2018), pp. 2405–2415.

[119] Samantha Tetef, Niranjan Govind, and Gerald T. Seidler. "Unsupervised Machine Learning for Unbiased Chemical Classification in X-ray Absorption Spectroscopy and X-ray Emission Spectroscopy". In: *Physical Chemistry Chemical Physics* 23.41 (Nov. 2021). DOI: 10.1039/D1CP02903G.

[120] Feelly Tumakaka, Joachim Gross, and Gabriele Sadowski. "Thermodynamic modeling of complex systems using PC-SAFT". In: *Fluid Phase Equilibria* 228-229 (Feb. 2005), pp. 89–98. DOI: 10.1016/j.fluid.2004.09.037.

[121] Cenk Ündey, Eric Tatara, and Ali Çınar. "Intelligent real-time performance monitoring and quality prediction for batch/fed-batch cultivations". In: *Journal of Biotechnology* 108.1 (2004), pp. 61–77.

[122] Oliver T Unke and Markus Meuwly. "PhysNet: a neural network for predicting energies, forces, dipole moments, and partial charges". In: *Journal of chemical theory and computation* 15.6 (2019), pp. 3678–3693.

[123] José O. Valderrama. "The State of the Cubic Equations of State". In: *Industrial & Engineering Chemistry Research* 42.8 (2003), pp. 1603–1618. ISSN: 0888-5885. DOI: 10.1021/ie020447b. URL: https://doi.org/10.1021/ie020447b.

[124] Luis Vaquerizo and María José Cocero. "CFD–Aspen Plus interconnection method. Improving thermodynamic modeling in computational fluid dynamic simulations". In: *Computers & Chemical Engineering* 113 (2018), pp. 152–161. ISSN: 0098-1354. DOI: https://doi.org/10.1016/j.compchemeng.2018.03.019. URL: https://www.sciencedirect.com/science/article/pii/S0098135418301959.

[125] Moritz Von Stosch et al. "Hybrid semi-parametric modeling in process systems engineering: Past, present and future". In: *Computers & Chemical Engineering* 60 (2014), pp. 86–101.

[126] W. Wagner and A. Pruß. "The IAPWS Formulation 1995 for the Thermodynamic Properties of Ordinary Water Substance for General and Scientific Use". In: *Journal of Physical and Chemical Reference Data* 31.2 (2002), pp. 387–535. DOI: 10.1063/1.1461829. eprint: https://doi.org/10.1063/1.1461829. URL: https://doi.org/10.1063/1.1461829.

[127] H. G. Weller et al. "A tensorial approach to computational continuum mechanics using object-oriented techniques". In: *Computers in Physics* 12.6 (1998), pp. 620–631. DOI: 10.1063/1.168744. eprint: https://aip.scitation.org/doi/pdf/10.1063/1.168744. URL: https://aip.scitation.org/doi/abs/10.1063/1.168744.

[128] Paul Westermann and Ralph Evins. "Adaptive Sampling For Building Simulation Surrogate Model Derivation Using The LOLA-Voronoi Algorithm". In: Jan. 2019, pp. 1559–1563. DOI: 10.26868/25222708.2019.211232.

[129] Tina R White. *A clustering algorithm for reduced order modeling of shock waves*. Tech. rep. Department of Mechanical Engineering, Stanford University, 2015.

[130] Jared Willard et al. "Integrating physics-based modeling with machine learning: A survey". In: *arXiv preprint arXiv:2003.04919* 1.1 (2020), pp. 1–34.

[131] Eric Winsberg. "Computer Simulations in Science". In: *Stanford Encyclopedia of Philosophy* (2019).

[132] Louise Wright and Stuart Davidson. "How to tell the difference between a model and a digital twin". In: *Advanced Modeling and Simulation in Engineering Sciences* 7.1 (2020), pp. 1–13.

[133] Shengli Xu et al. "A Robust Error-Pursuing Sequential Sampling Approach for Global Metamodeling Based on Voronoi Diagram and Cross Validation". In: *Journal of Mechanical Design* 136 (July 2014), p. 071009. DOI: 10.1115/1.4027161.

[134] Zhuo Yang et al. "Investigating Grey-Box Modeling for Predictive Analytics in Smart Manufacturing". In: Aug. 2017, V02BT03A024. DOI: 10.1115/DETC2017-67794.

[135] Kun Yao et al. "The TensorMol-0.1 model chemistry: a neural network augmented with long-range physics". In: *Chemical science* 9.8 (2018), pp. 2261–2269.

[136] Jie Yu, Kuilin Chen, and Mudassir M Rashid. "A Bayesian model averaging based multi-kernel Gaussian process regression framework for

nonlinear state estimation and quality prediction of multiphase batch processes with transient dynamics and uncertainty". In: *Chemical Engineering Science* 93 (2013), pp. 96–109.

[137] Valentina Zambrano et al. "TWINKLE: A digital-twin-building kernel for real-time computer-aided engineering". In: *SoftwareX* 11 (2020), p. 100419.

[138] Sohrab Zendehboudi, Nima Rezaei, and Ali Lohi. "Applications of hybrid models in chemical, petroleum, and energy systems: A systematic review". In: *Applied energy* 228 (2018), pp. 2539–2566.