PhD-FSTM-2023-065
The Faculty of Science, Technology and Medicine

DISSERTATION

Defence held on 29/06/2023 in Esch-sur-Alzette

to obtain the degree of

DOCTEUR DE L'UNIVERSITÉ DU LUXEMBOURG

EN SCIENCES DE L'INGÉNIEUR

by

Fateme Darlik
Born on 2nd April 1994 in Booshehr, Iran

PHYSICALLY INFORMED NEURAL NETWORKS TO REPRESENT MOTION OF GRANULAR MATERIAL

Dissertation defence committee

Dr Bernhard Peters, dissertation supervisor
Professor, Université du Luxembourg

Dr Davide Baroli
Researcher, Università della Svizzera italiana

Dr Jakub Lengiewicz
Professor, Technische Universität München

Dr Andreas Zilian, Chairman
Professor, Université du Luxembourg

Dr Viktor Scherer
Professor, Ruhr-Universität Bochum

Affidavit

I hereby confirm that the PhD thesis entitled Physically informed neural networks to represent motion of granular material has been written independently and without any other sources than cited.

Luxembourg

Fateme Darlik

Name

Dedications

I would like to dedicate this thesis:

To my amazing husband, who is not only my partner in life but also my best friend. Your support and understanding during the challenging moments have kept me motivated and focused. This achievement would not have been possible without your constant belief in me and your willingness to lend a helping hand whenever needed.

To my family, whose love and support have been the driving force behind my pursuit of knowledge and academic achievements.

To my advisor, whose expertise and guidance have shaped me into a better researcher and have inspired me to push the boundaries of my knowledge.

To my friends and colleagues who have walked alongside me on this academic journey. Your insightful discussions and shared experiences have made this journey not only intellectually stimulating but also enjoyable.

To all those who have played a part in my academic journey, directly or indirectly.

Summary

The growing importance of renewable energy sources, particularly biomass, in mitigating climate change has led to increased research and development in this field. Biomass combustion chambers play a crucial role in converting biomass into heat energy efficiently and cleanly. However, the combustion process is influenced by the characteristics of biomass particles, such as their composition, size distribution, and moisture content, which can vary significantly. In this thesis, we explore the application of physics-informed neural networks (PINNs) for predicting particle motion in biomass systems, aiming to improve combustion efficiency and reduce emissions.

The thesis is divided into three main parts. The first part introduces a PINN model for predicting biomass particle velocities in a moving bed. The proposed model demonstrates the effectiveness of PINNs as surrogate models for computationally expensive discrete element method (DEM) simulations. The second part focuses on reconstructing biomass particle fields in a particle-fluid problem using continuum methods. Through the integration of simulation data and PINNs, accurate predictions of particle velocities, pressure, and density fields are achieved. In the third part, a recurrent neural network architecture is employed to predict particle motion in a moving grate chamber and a rotating drum. The results highlight the model's ability to capture particle behavior accurately, even for time intervals longer than the training time.

The findings of this thesis contribute to the understanding of biomass combustion dynamics and offer valuable insights into optimizing biomass combustion chamber design and operation. The PINN-based approach proves to be computationally efficient and reliable for predicting particle motion, providing a valuable tool for biomass system analysis and optimization. Additionally, a separate study on three-dimensional computational fluid dynamics - discrete element method (CFD-DEM) simulations of raceway transport phenomena in a blast furnace is presented, emphasizing the importance of understanding the complex dynamics in metallurgical processes.

By combining the power of neural networks with physics-based modeling, this research

bridges the gap between computational efficiency and accuracy in predicting particle motion in biomass systems. These insights can contribute to the development of cleaner and more efficient biomass combustion technologies, promoting a sustainable and greener future.

Contents

[1] Fateme Darlik, Prasad Adhav, and Bernhard Peters. "Prediction of the biomass particles through the physics informed neural network." In: *ECCOMAS Congress 2022-8th European Congress on Computational Methods in Applied Sciences and Engineering*. 2022.

[2] Fateme Darlik and Bernhard Peters. "Reconstruct the biomass particles fields in the particle-fluid problem using continuum methods by applying the physics-informed neural network". In: *Results in Engineering* 17 (2023), p. 100917.

[3] Fateme Darlik and Bernhard Peters. "Using the trained recurrent neural network to predict particles position for time more than training time". In: (Sept. 2023). Unpublished Manuscript.

[4] Navid Aminnia, Prasad Adhav, Fateme Darlik, Muhammad Mashhood, Sina Hassanzadeh Saraei, Xavier Besseron, and Bernhard Peters. "Three-dimensional CFD-DEM simulation of raceway transport phenomena in a blast furnace". In: *Fuel* 334 (2023), p. 126574.

List of Figures

List of Tables

Chapter 1

Introduction

In recent times, there has been a growing acknowledgment of the importance of renewable energy sources in worldwide energy, aiming to reduce the detrimental impacts of climate change. Biomass, as a versatile source of renewable energy, has gained significant attention [1]. The utilization of biomass not only offers a sustainable alternative to fossil fuels but also presents opportunities for reducing greenhouse gas emissions and promoting a cleaner and greener future.

Biomass refers to organic matter derived from living or recently living organisms [2], such as plants, agricultural waste, forestry residues, and dedicated energy crops (figure 1.1). It is an important resource that can be transformed into energy via a variety of procedures. One of the most important applications of biomass is in combustion, which is the thermal conversion of biomass into heat energy.

A biomass combustion chamber is crucial in this process. It aims to ignite biomass sources effectively, transforming their chemical energy into heat. The combustion chamber generates a controlled atmosphere in which the biomass fuel burns, providing heat that can be used in mechanical applications [4]. The combustion chamber's design and operation are critical for attaining optimal combustion efficiency, minimizing emissions, and assuring safe and reliable operation.

Biomass power plants generate steam in combustion chambers, which power turbines to

Figure 1.1: Biomass Sources - Illustration depicting various biomass sources, including plants, agricultural waste, forestry residues, and energy crops, highlighting their importance as renewable energy feedstocks [3].

generate energy. Because it provides a clean and renewable energy solution, this application has tremendous promise, particularly in regions with substantial biomass supplies.

Furthermore, biomass combustion chambers are employed in industrial processes requiring high-temperature heat, such as building heating systems, industrial furnaces, and drying applications [4]. Biomass combustion can provide a dependable and cost-effective heat source, providing an alternative to fossil-fuel-based systems.

In certain mechanical applications, such as the production of biofuels, biomass has been processed before entering the combustion chamber. The mixing drum procedure is an important one in this sense. The mixing drum is a mechanical device that is used to homogenize biomass feedstocks to ensure uniform composition and qualities. It makes it easier to combine different biomass sources, improving fuel quality and combustion characteristics.

Adequate mixing of biomass sources in the mixing drum considerably leads to fuel variability reduction. Depending on the processing method, biomass fuels can vary in moisture content, particle size distribution, and chemical composition. These changes can have an instantaneous impact on the combustion process, resulting in incomplete combustion, and greater emissions.

The variety in fuel qualities can be reduced by putting biomass sources into the mixing drum process. The drum aids in the homogenization of various biomass sources, ensuring that the fuel mixture has a uniform composition [6]. This uniformity not only increases combustion efficiency but also minimizes risks of incomplete combustion, and pollutant emissions.

Furthermore, the uniformity achieved through the mixing drum process has a direct impact on the overall system performance [7]. Consistent fuel properties enable better control and optimization of the combustion chamber's design and operation. This optimization includes factors such as air-fuel ratio, residence time, and temperature distribution, which are crucial for achieving maximum combustion efficiency and minimizing operational issues.

Machine learning encompasses various approaches and techniques that enable computers

Figure 1.2: Mixing drum: With a smooth rolling action, the drum can ensure efficient mixing of the materials inside it [5].

to learn from data and make predictions or decisions without being explicitly programmed. Among the different types of machine learning, neural networks have gained significant attention due to their ability to model complex relationships and patterns.

Neural networks are computational models inspired by the structure and functioning of the human brain. They consist of interconnected nodes, called neurons, organized in layers. Each neuron receives inputs, applies a mathematical transformation to them, and produces an output. By adjusting the weights and biases associated with these connections, neural networks can learn from data and make predictions or classifications.

One of the main advantages of neural networks is their ability to handle complex and non-linear relationships in the data. They can capture intricate patterns and representations that may not be easily discernible using traditional statistical or rule-based approaches. Neural networks excel in tasks such as image recognition, natural language processing, and speech recognition, where the underlying patterns are complex and multidimensional.

Another advantage of neural networks is their capability to perform feature extraction automatically. Traditional machine learning algorithms often require handcrafted feature

engineering, which is a time-consuming and knowledge-intensive process. Neural networks, on the other hand, can learn relevant features directly from the raw data, reducing the need for manual feature engineering.

Neural networks, however, also have some limitations. One notable disadvantage is their black-box nature, meaning that it can be challenging to interpret and understand how they arrive at their predictions. This lack of interpretability can be a significant concern, especially in critical applications such as healthcare or finance, where transparency and interpretability are crucial.

Furthermore, neural networks are computationally expensive and require large amounts of data for training. The training process involves iterative optimization algorithms that adjust the weights and biases of the network to minimize a loss function. This process can be time-consuming and computationally intensive, particularly for deep neural networks with many layers and parameters.

In the context of mechanical applications, where understanding the underlying physics is essential, physics-informed neural networks (PINNs) have emerged as a powerful approach. PINNs combine the strengths of neural networks and physics-based modeling, incorporating the governing equations of the problem into the neural network's loss function.

The advantage of PINNs is that they enforce physical constraints on the neural network's predictions, enabling better regularization and utilization of the underlying physics. By incorporating the physics equations as part of the loss function, PINNs provide a mechanism for incorporating prior knowledge and constraints, improving the accuracy and stability of the predictions.

The proposed methodology combines the strengths of Computational Fluid Dynamics (CFD) and Discrete Element Method (DEM) in an Eulerian-Lagrangian multiphase model. The transport equations are solved for the continuous fluid phase, while the particles are treated as discrete entities using the Lagrangian approach of DEM. The interaction between the fluid and particles is accounted for through mass, momentum, and energy exchange. The discrete element method allows for the modeling of particle movement, rotation, and

interaction. Traditional approaches, such as Computational Fluid Dynamics-Discrete Element Method (CFD-DEM), have limitations when it comes to dealing with complex geometries and an enormous amount of particles. Neural networks have been applied to predict hydrodynamic properties, drying solids in fluidized beds, and biomass gasification to tackle these restrictions. However, neural network models, although computationally efficient, often lack a strong sense of physics in fluid dynamics problems. To address this limitation, physics-informed neural networks have been developed. These models incorporate physical principles by including the governing equations as part of the neural network's loss function, leading to better regularization and utilization of the physics behind the problem [8].

Physics-informed neural networks have shown promise in predicting particle behavior in fluidized beds without requiring explicit information about boundary conditions, initial conditions, or geometry. This approach has proven effective in solving inverse problems and data assimilation, as well as utilizing the hidden physics of multi-physics problems [9, 10, 11].

The physics-informed neural networks have been used widely especially in the field of particle-fluid dynamics, mainly when data production via simulation is computationally expensive or only sparse experimental data is available [8].

To solve the governing equations of the problem, a physics-informed neural network is employed. The neural network takes the input of time, and position and predicts the variables of interest such as velocity, density, and pressure. The use of neural networks allows for efficient and accurate predictions, avoiding the need for coupling CFD and DEM methods.

Figure 1.3 illustrates a schematic representation of a basic neural network, denoted by the red dashed line on the left side. This neural network receives input variables t, x, and y and produces corresponding outputs u and v. In the context of Physics-Informed Neural Networks (PINNs), if the relationship between the input variables (x) and the output variables (u) can be described by partial differential equations e_1, e_2 and e_3, these equations

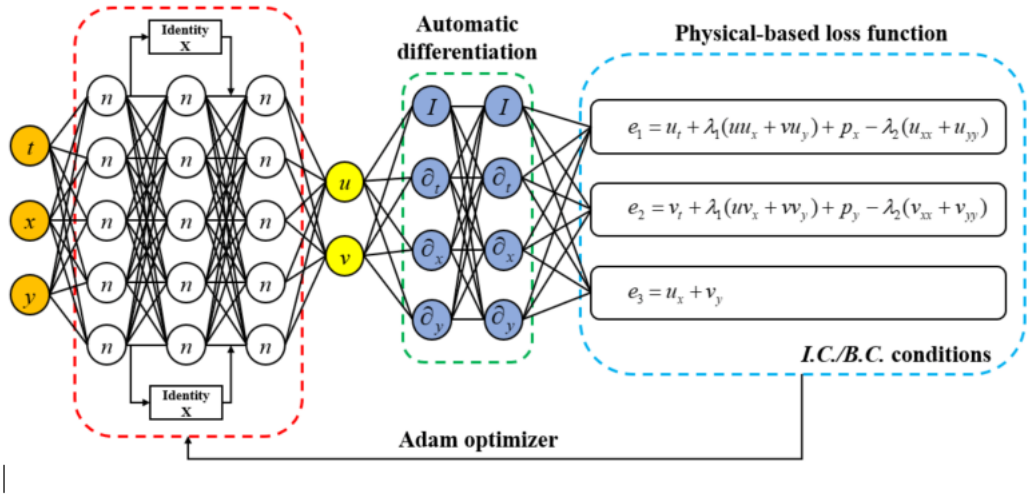


Figure 1.3: Physics-Informed Neural Networks (PINNs) incorporating the governing equations of the problem into the neural network's loss function, allowing for a better understanding of the underlying physics [12].

can be incorporated into the loss function.

Once the neural network calculates the output values, it not only checks the proximity of the outputs to the exact values but also verifies if they satisfy the underlying partial differential equations. This evaluation is achieved through automatic differentiation, which allows for the calculation of derivatives within the network. Consequently, the loss function is used to adjust the weights and biases of the neural network, aiming to improve its performance in approximating the desired outputs and adhering to the specified partial differential equations. The thesis consists of three main parts, each focusing on the application of a physics-informed neural network (PINN) for predicting particle motion in different systems.

The first part introduces a paper titled "Prediction of the biomass particles through the physics-informed neural network [13]". It addresses the challenge of simulating the motion of biomass particles, which is computationally expensive due to the large number of particles and the required simulation time. The proposed PINN model incorporates the equations of motion of the particles, enabling the reconstruction of velocity fields and

reducing processing effort. The paper demonstrates the effectiveness of PINNs as surrogate models for computationally costly discrete element method (DEM) simulations. It provides an overview of biomass furnaces, the use of DEM simulations, and the data generation process using the XDEM software. The PINN methodology is explained, and the results show reliable prediction of particle velocities in a moving bed.

The second paper "Reconstruct the biomass particles fields in the particle-fluid problem using continuum methods by applying the physics-informed neural network [14]" focuses on the aim of analyzing the fluid-like behavior of dense particles in a moving grate combustion chamber. Due to the lack of access to experimental datasets, a simulation approach is used, employing the XDEM software. The researchers collect datasets of particle positions, velocities, and interaction forces and incorporate them into a PINN. Hyperparameter tuning is performed to optimize the neural network's performance, and the effects of changing the number of hidden layers and neurons per layer are investigated. The trained network accurately predicts velocities, pressure, and density fields, with good agreement with simulation results.

The third paper "Using the trained recurrent neural network to predict particle position for time more than training time" presents the results and discussions of two case studies utilizing a neural network model. In the first case study, a two-network architecture is proposed to predict particle motion in a moving grate chamber. The neural networks are trained on a dataset generated from a simulation, and the results demonstrate low prediction errors and plausible particle behavior. In the second case study, the neural network model is applied to the mixing process of particles in a rotating drum. The network's performance evaluation shows accurate predictions of particle velocities and positions, indicating its effectiveness in capturing particle motion patterns.

Overall, the above chapters show the application of PINNs in predicting particle motion in various systems. The results demonstrate the reliability, computational efficiency, and accuracy of the proposed neural network model, offering valuable insights into particle dynamics.

The last chapter, "Three-dimensional CFD-DEM simulation of raceway transport phenomena in a blast furnace [15]", highlights the importance of simulating the raceway phenomenon in a blast furnace using a 3D CFD-DEM model. The results demonstrate that the 3D model provides a better approximation of the raceway and accurately captures the transport of species, dynamics, and temperature distribution. The discretization of particles enables the model to estimate temperature gradients and accurately calculate reaction rates. The study focuses on coke combustion but suggests that further research should include iron particles to analyze oxidation reactions and phase changes in the cohesive zone of the blast furnace. By extending the current raceway model, it is possible to gain more detailed insights into blast furnace operations and contribute to reducing costs and carbon emissions in this important metallurgical process.

Chapter 2

Prediction of the biomass particles through the physics-informed neural network[1]

[1] Fateme Darlik, Prasad Adhav, and Bernhard Peters. "Prediction of the biomass particles through the physics informed neural network." In: *ECCOMAS Congress 2022-8th European Congress on Computational Methods in Applied Sciences and Engineering*. 2022.

Abstract

Woody biomass energy is a kind of renewable energy that contributes to the reduction of greenhouse gas emissions, the creation of healthier forests, and the reduction of wildfire danger. Generally speaking, simulations of the motion of biomass particles are a time consuming process due to a large number of particles and required simulation time. We used a physics-informed neural network (PINN) model to predict the motion of particles by including their equations of motion to reconstruct the velocity fields and reduce the processing effort. compare to the discrete element methods, the PINNs methods have the advantage of predicting the velocity fields without the knowledge of the simulation's boundary and initial conditions as well as geometry. It has shown that the proposed model has reliable prediction results with a mean percentage error in time less than 1 percent.

Keywords: Physics informed neural network, Discrete element method, Biomass furnace, Surrogate model

2.1 Introduction

A biomass furnace uses a well known process of gasification to turn biomass into flue gases. These flue gases are then in turn used to generate heat, which is used for different purposes, usually to generate steam and run steam turbines to generate electricity. In the day and age of climate change, biomass furnace is catching a lot of attention as an alternative renewable energy source [16]. The fuels often used are high in moisture content, a variety of minor constituents such as chlorine, sulfur, phosphorous, and a variety of ash-forming metals [17]. Hence, it is essential to study and predict these operating parameters for different fuels, especially new fuels such as domestic waste. These parameters are heavily dependent on the conversion of the fuel into flue gases and the burning of the flue gases. As performing experiments for biomass simulation is prohibitively expensive and time-consuming, numerical simulations are used to solve the problem.

The Discrete Element Method (DEM) is used to simulate the biomass moving bed. The

DEM treats biomass fuels as particles. The DEM handles particle motion, interparticle collisions, and solid-state reactions. Such a numerical model of the biomass furnace contains thousands of particles. These particles then travel across the biomass furnace using fixed and moving grates. The numerical simulation for such particle movement is computationally costly, especially when a large number of particles for a long simulation time is used. Hence this becomes a major hurdle when modeling biomass furnaces with long grates and small size fuel "particles". For a given biomass furnace design, the particle bed movement does not change much for a given size and shape. The composition of the particle bed can surely change, which is the interesting part when considering the biomass furnace operating parameters (apart from the particle bed movement). Thus it could be very beneficial if this particle bed movement could be predicted in a computationally cheaper way. In the present study, Physics-Informed Neural Networks (PINNs) are employed to predict the moving particle bed motion and are intended to be used as a surrogate model for the computationally costly DEM model. Neural network (NN) models are increasingly being used in a wide range of areas since they are effective tools for resolving, improving, and optimizing scientific problems. Neural networks can understand the implicit, complicated and vague information and based on that, are able to recognize the pattern behind it and predict the assigned outputs. Altogether, the neural network is proven to be a useful tool in solving engineering problems [18], especially in scenarios where there is not sufficient data to use for doing simulation [19], the simulation is time-consuming [20], and where optimization, decision making, and data analysis are the concerns.

Generally speaking, there are different types of neural networks such as fully connected neural networks [21], feed-forward artificial neural networks [22], convolution neural networks [23], recurrent neural networks [24], and physics-informed neural networks (PINNs) [25] which are based on the need of the problem. Among all of these, in this study, the PINNs are used as neural network tools. The main advantage of the PINNs [26], is that they are more suitable for the physical concerns of the engineering problems and their governing equation [27]. In contrast, when using a non-physics-informed neural network,

the existence of a sufficient number of data is mandatory to have a well-trained neural network, however, in PINNs, the govern equations of the problem will give the neural network a better understanding of the problem itself [28]. Following that, PINNs are practical in problems where data generation via simulation is computationally expensive, or the only data available are the sparse data from experimental tests [29, 30]. PINNs are aware of data limitations and will predict the outputs by using extensive physical knowledge of the problem. Additionally, PINNs can be used with or without adding the information regarding boundary conditions, initial conditions [31, 32], and the geometry [33, 34], which should be decided based on the user concerns.

The motion of the particles in the moving bed is extracted from simulation data in this study. The PINN is constructed by adding the motion equations as the governing equation to the neural network. The PINN predicts and reconstructs the velocity fields of particles moving on moving grate beds while taking into account unexpected changes in the particles' downward motion caused by gravity. It is demonstrated here that PINNs are effective instruments capable of predicting the motion fields. Additionally, the time-average percent error of the velocity does not exceed one percent error, which explains the capability of our neural network. The paper is organized as follows: In section 2.2 we present the mathematical modeling of DEM and PINN training data generation. In section 2.3, the methodology for developing the PINN is described. In section 2.4 we present the results from the PINN. Finally, in section 2.5 we conclude the findings of the paper.

2.2 DEM Simulation for data generation

Generally, real-world data or data based on the real world is used to train a NN. In the presented work, we use the data obtained from numerical simulations of the Biomass furnaces using the eXtended Discrete Element Method (XDEM) [35] software. XDEM software is based on the Discrete Element Method (DEM), which can simulate particle motion, interparticle collisions, etc. This DEM software is then extended to account for

inter-particle and intra-particle thermodynamics as well as the influence of fluid flow. XDEM is thoroughly verified and validated against experimental studies [36, 37, 38, 39, 40, 35], including validation for biomass furnace and processes involved in it. The mathematical modeling of XDEM is presented in the following sections.

The dynamics module of XDEM software predictions particle position, velocity, and acceleration. It uses a soft sphere model, where particles are assumed to be deformable and can overlap each other. The inter-particle contact forces are computed through the magnitude of overlap between the particles, based on the force-displacement law chosen. The particle hardness is described with Young's modulus, whereas the particle energy dissipation is modeled with a dampener and/or dash-pot. The transnational and rotational movements of individual particles are tracked using the classical mechanics equations as described in Eq (5.1) and Eq (5.2) respectively.

$$m_i \frac{d\vec{v_i}}{dt} = m_i \frac{d^2\vec{X}_i}{dt^2} = \vec{F_i^c} + \vec{F_i^g} + \vec{F_i^{ext}} \tag{2.1}$$

$$I_i \frac{d\vec{\omega_i}}{dt} = \sum_{j=1}^{n} \vec{M}_{i,j} \tag{2.2}$$

where $\vec{F_i^{ext}}$ is the sum of all the external forces acting on the particle, such as particle impact forces, buoyancy forces $\vec{F_B}$ and drag forces $\vec{F_D}$. The inter-particle impact energy dissipation is modeled as a spring and dashpot.

2.2.1 Biomass furnace simulation set up

In the present study, the dynamics module of the XDEM software is used to simulate a 2D moving bed in the combustion chamber of a biomass furnace. The biomass fuel is modeled as spherical particles. The biomass furnace case is set up as shown in figure 4.1. The moving bed composition consists of Beechwood and Stringy wood (agricultural waste) with properties described in table 2.1. The biomass fuel is fed into the furnaces through the 2 particle sources (for each type of fuel) from the left as shown in the figure 4.1.

Table 2.1: Characteristics of Particles: Properties of Beechwood and Stringy-wood

Properties	Beechwood	Stringy-wood
Particle Radius [m]	0.015	0.015
Particle bed mix fraction [%]	79.5	20.5
Mass Flow rate [kg/s]	0.0155	0.0038
Number of particles	1762	462
Apparent Density [kg/m^3]	648.52	610.47
Mean porosity [-]	0.685196	0.6375

The biomass furnace consists of grates, fixed (black) and moving (red). These moving grates are used to move the particles along the length of the biomass furnace. As time progresses the particles will eventually fall from the outlet opening. Partial walls of the biomass furnace are added to show a general outline of the furnace.

2.2.2 XDEM Results

The XDEM simulation is run from 0 s to 100 s. As seen in figure 2.2, we can see the moving (red) grates pushing particles along. This grate movement dictates the traverse speed of particles along the length. This speed in turn then dictates the residence time of particles in particular zones of the furnace (such as drying, pyrolysis, char burning, etc.). The moving bed comes into regular motion after around 40 s, after which the moving bed motion does not vary much. Considering this the particle position and velocities between 40 s and 100 s are used as training data for the PINN.

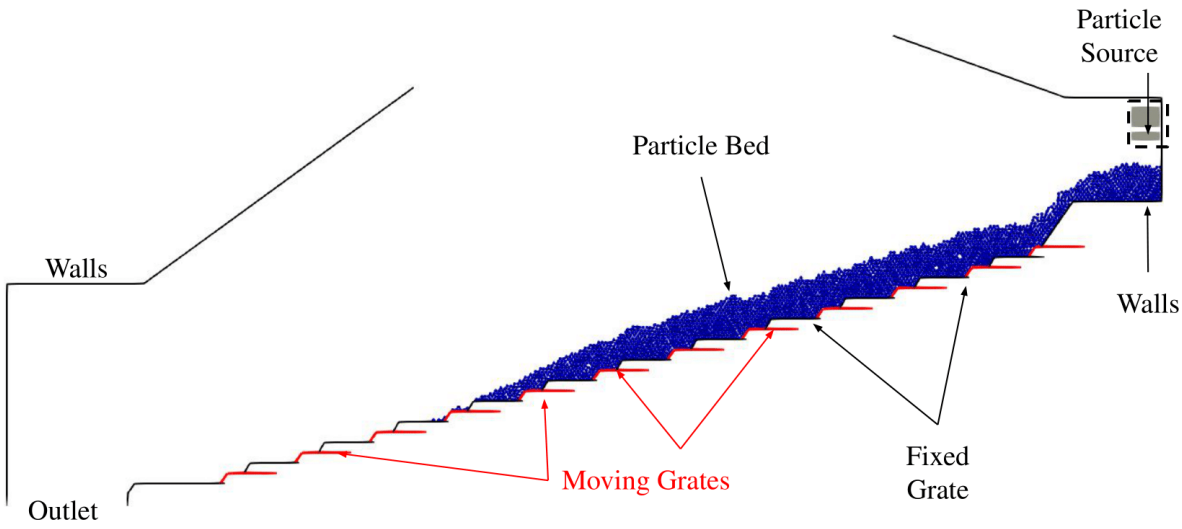


Figure 2.1: XDEM Biomass furnace Case setup

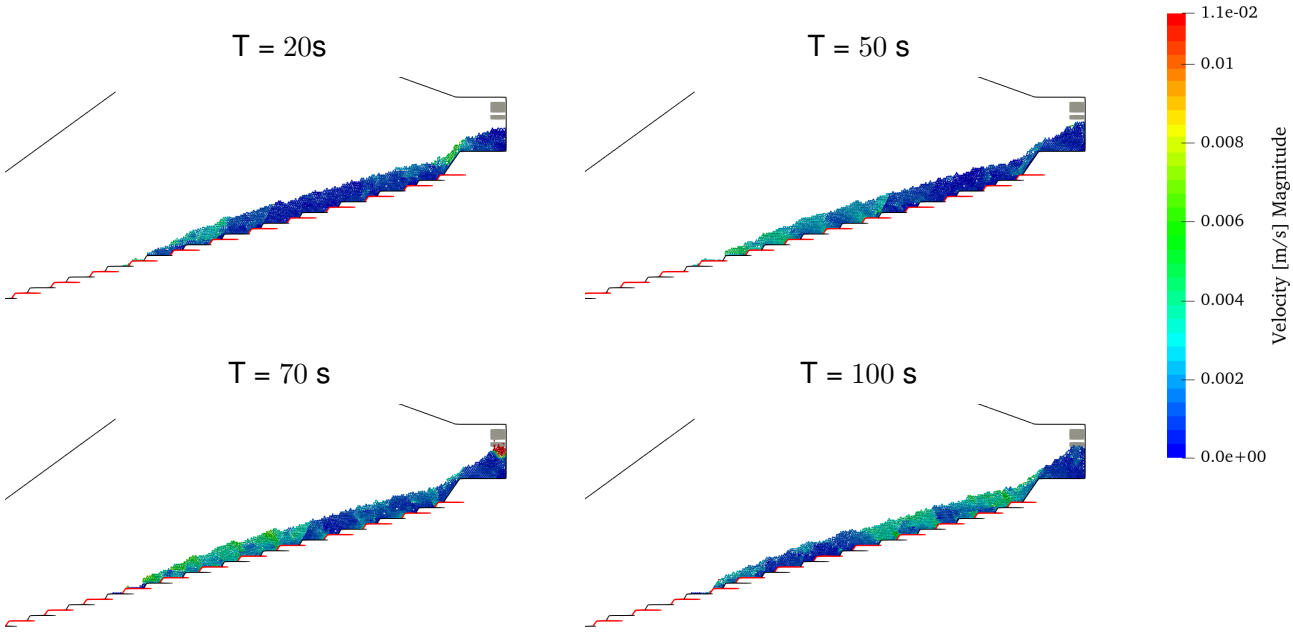


Figure 2.2: XDEM simulation results showing the progression of moving bed in Biomass furnace

2.3 Physics-Informed neural network

PINNs [25] is an extended type of scientific neural network that takes model equations such as partial differential equations (PDEs) into account, as part of its neural network. Nowadays, the application of PINNs [27]is expanding due to their efficiency in satisfying the equations in both supervised and unsupervised learning. It can approximate the solution of PDEs equations while training the neural network. Additionally, by using this methodology one can evaluate the boundary conditions and initial conditions needed for solving PDEs [29]. In other words, PINNs are categorized in deep learning networks that can produce an approximate solution in the integration domain for each point included in the PDEs equations. The residuals of PDEs are included in the loss function of its neural network which will directly affect the efficiency of the process. Another main advantage of the PINNs is that they can be a mesh-free methodology that can solve the PDEs by using a direct problem solution and changing it to the optimization problem. It reinforces the loss function by residual terms of the governing equations which can be interpreted as a penalizing term that forces the output of the neural network to be in an acceptable solution. Generally speaking by using this framework, the neural network not only predicts the outputs but also checks if the predicted variables can satisfy the governing equations [27]. It has proven to be practical, especially in mechanical engineering data-driven studies. PINNs consider underlying PDE which introduces the physics of the problem rather than just predicting the solution only based on the data, by reducing the error between the predicted data and actual data. One of the areas in that PINNs has been used widely is in the inverse problem to reconstruct the fluid field by using the data from the sensors [27]. By feeding the neural network the sparse data that is available from one domain, and adding the governing equations to the neural network, the PINNs can reconstruct the domain fields. This approach has been implemented in different fluid dynamics problems [41]. In this study, we aim to use the PINN to reconstruct the velocity fields for particles moving in the moving beds of fluids in the two-dimensional domain. In the figure 2.3 the architecture of the physics-informed neural network is depicted. The equations of the motion are added to the loss function to assure

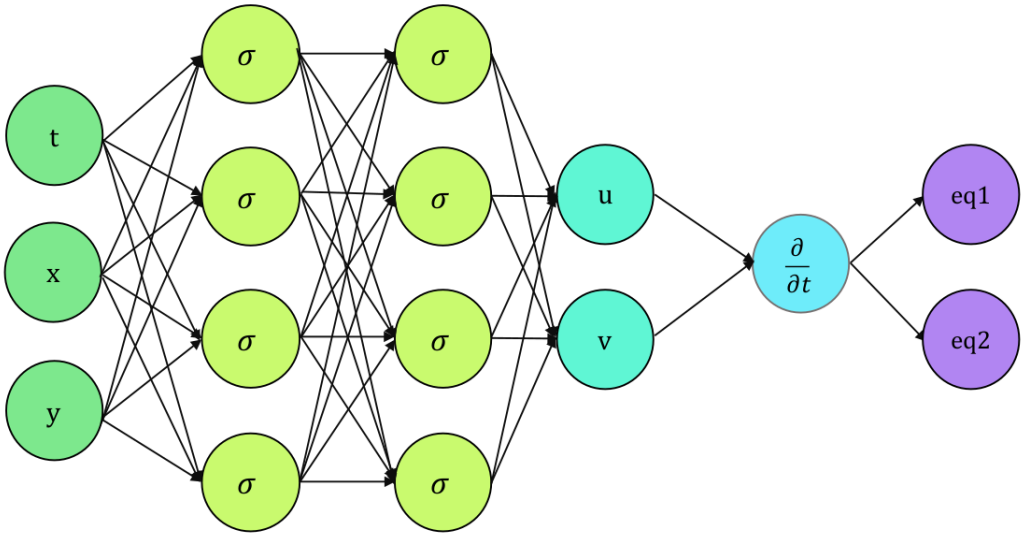


Figure 2.3: Physics-Informed neural network architecture. The inputs are the coordinates (x and y) and the simulation time. The output is the velocity in the x and y direction. The governing equations are the equation of the motion.

the neural not only predicts the output data but takes care of the fact that the output data should satisfy the equations.

2.4 Results and discussion

The PINN methodology is used to predict the velocity of the particles in a moving bed. The combustion chamber case study that is used in this article is thoroughly detailed in the section 2.2 above. Utilizing XDEM, the simulation is done, and the required data is then collected for usage as train and test data to feed the neural network. After fine-tuning the neural network's hyperparameters, a network with 10 layers and 200 neurons per layer is employed, yielding 323203 parameters. The activation function rectified linear unit (ReLU) (2.3) is applied for the hidden layer and the activation function sigmoid (2.4) is used for the output layer.

$$ReLU(x) := max(0, x) \tag{2.3}$$

$$\sigma(x) := \frac{1}{1 + e^{-x}} \tag{2.4}$$

The simulation time steps and the coordinates are used as the input of the neural network, and the velocities are the output of the neural network. As the order of the magnitude of the data is different from each other, a normalization between 0 to 1 is applied to all data including inputs and outputs. The mean squared error (MSE) is used for the loss function. The Adam optimization method is chosen as the optimization method as it is a stochastic gradient descent approach that is highly efficient, requires low memory, and is appropriate for large data problems [42]. An adaptive learning rate that started from $1e-3$ and decreases to $1e-5$ is used to control how quickly the model is adapted to the problem. The batch size is chosen to be equal to the total number of particles in each time step. To avoid overfitting, the total number of epochs is not defined from the start; training of the neural network continues until we get a minimum for the validation dataset. Of the whole number of data, 70 percent of it randomly is chosen as the training dataset, and the other 30 percent is considered as the test data set. And 10 percent of the training dataset is used as the validation dataset. After training for about 20,000 epochs, and monitoring the changes in the loss function, we stopped the training.

The equations that are added to neural networks are the equations of the motion (equations (2.5) and (2.6)) in the both x and y direction.

$$e_1 := f_x(t, x, y) - m\frac{du(t, x, y)}{dt} \tag{2.5}$$

$$e_2 := f_y(t, x, y) - m\frac{dv(t, x, y)}{dt} \tag{2.6}$$

Where $u(t, x, y)$ is the particle velocity in x direction, $v(t, x, y)$ is the particle velocity in y direction, m is particles mass, and $f_x(t, x, y)$ and $f_y(t, x, y)$ are the forces applied on the particles in x and y direction respectively. The particles' applied forces are consists of interaction force, and the weights of the particles are all calculated by using the XDEM simulation. The mentioned equations are added to the PINN's loss function.

$$MSE := \sum_{n=1}^{N} \Big(|u(t^n, x^n, y^n) - \hat{u}^n|^2 + |v(t^v, x^n, y^n) - \hat{v}^n|^2 + \sum_{i=1}^{2} \sum_{n=1}^{N} (|e_i(t^n, x^n, y^n)|^2) \tag{2.7}$$

Where N is the total number of the particles, $\hat{u}$ and $\hat{v}$ are the learned velocity. In order to illustrate the effectiveness of our neural network, the time mean squared error (equation (2.8)), and time mean squared error (equation (2.9)) are reported for each output field.

$$MSE^t(u_i^t, \hat{u}_i^t) := \frac{1}{N} \sum_{i=1}^{N} (u_i^t - \hat{u}_i^t)^2 \tag{2.8}$$

$$MRPE^t(u_i^t, \hat{u}_i^t) := \frac{1}{N} \sum_{i=1}^{N} \frac{u_i^t - \hat{u}_i^t}{u_i^t} \times 100 \tag{2.9}$$

MSE^t calculates the average squared error between the exact and predicted data in each time step t, and the $MRPE^t$ calculates the relative percentage error between the exact and he predicted data for all the available particles in each time step t. Figure 2.4 shows a diagram of the MSE values over time. With the results obtained within the plot, the maximum value is less than 10^{-3}, the average value for the velocity in both directions is 10^{-6}, and the majority of the errors are some number between 10^{-8} and 10^{-6}, which altogether represents a good agreement between the predicted data and the exact data. Another intriguing result is depicted in figure 2.5. It can be seen from the chart that the proposed framework is capable of reconstructing the velocity fields with a percent error of less than 1 present. By having a closer look at the figures 2.4 and 2.5, one anticipated finding will be the fact that while the error for the velocity in the y direction has a lower value compared to the error for the velocity in the y direction, however, the maximum error comes from the velocity in x direction. We can attribute this to the fact that the existence of gravity

in the y direction will lead to sudden changes in the value f the velocity which the neural networks would not be able to capture. On the other hand, if the particle is on the grates the velocity in the y direction are mainly the same as each other which will make it easier for the neural network to anticipate the behavior of the particles in these directions.

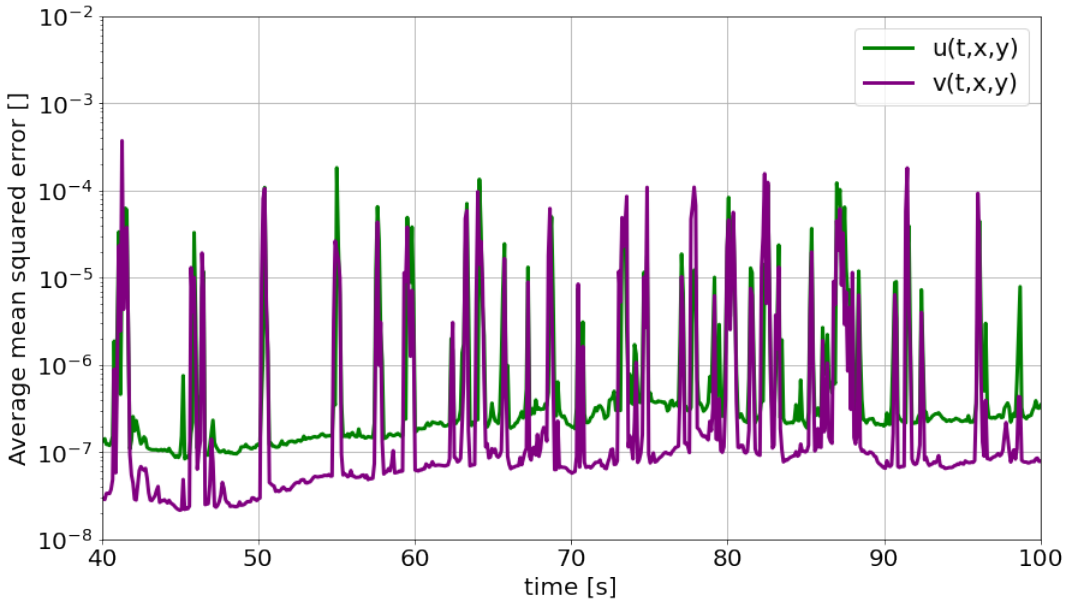


Figure 2.4: The average of the mean squared error of all the particles available in each time step.

After a detailed study of the figures 2.4 and 2.5, we find out that at time 50 seconds, the maximum error value for both plots happened at time 50 seconds. To validate our neural network, the representative snapshots of the exact data, learned data, and the relative percent error at a time of 50 seconds are depicted in figure 2.6. The first row of the figure 2.6 presents the contour of the exact value of velocity in both x and y directions that are derived from the DEM simulation. In the next row, the velocity predicted by PINNs is shown. Additionally, the last row shows the percent relative error between exact and predicted data which is calculated as equation (2.10):

$$RPE(u_i, \hat{u}_i) := \frac{u_i - \hat{u}_i}{u_i} \times 100 \tag{2.10}$$

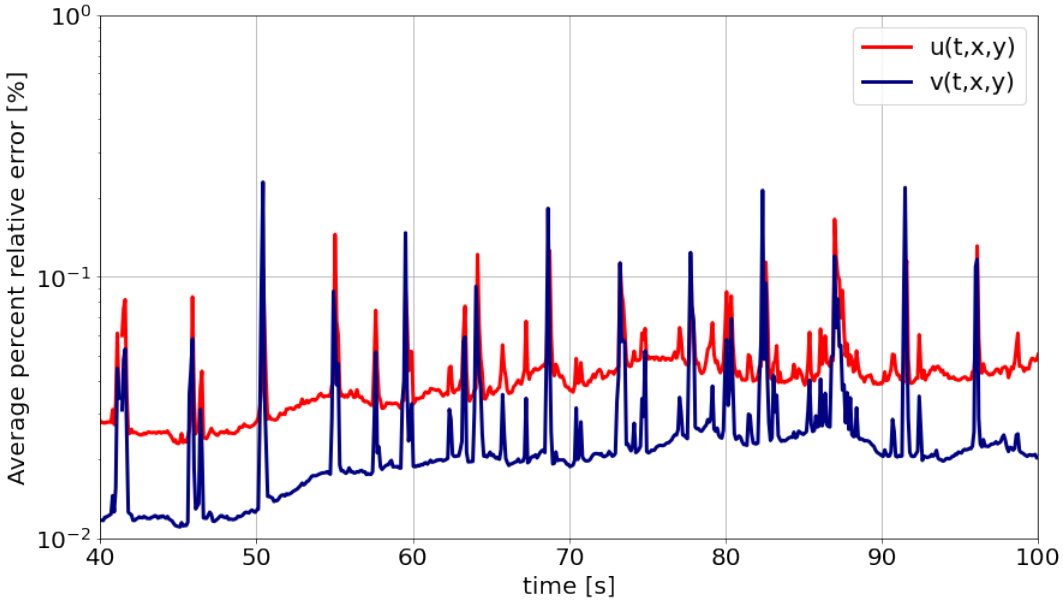


Figure 2.5: The average of the percentage error of all the particles available in each time step.

Looking at figure 2.6 demonstrates that when comparing the predicted results to the exact results, the neural network will give us a smother change in the velocity. The present frameworks are capable of predicting the velocity with a maximum percent error of 0.35 percent in the x direction and 0.16 in the y direction. As these chosen time steps have maximum MRPE compared to other time steps, it is expected that the neural network in the other time steps will predict with higher accuracy.

To get more detail about the performance of the used methodology the percentage error between the exact learned velocity for three different time steps is represented in figure 2.7. The time 50, 70, and 100 seconds is chosen to be used in this section. One anticipated finding is that the value of the error in the edge of grates is generally larger than in other parts of the simulation. Additionally, in the inlet of the domain, where we are having mass flows, due to lack of enough information, it consists of a larger value of error when compares to the middle part of the simulation. The same behavior can be seen in the outlet of the domain, And the explanation is that in these sections as the particles are

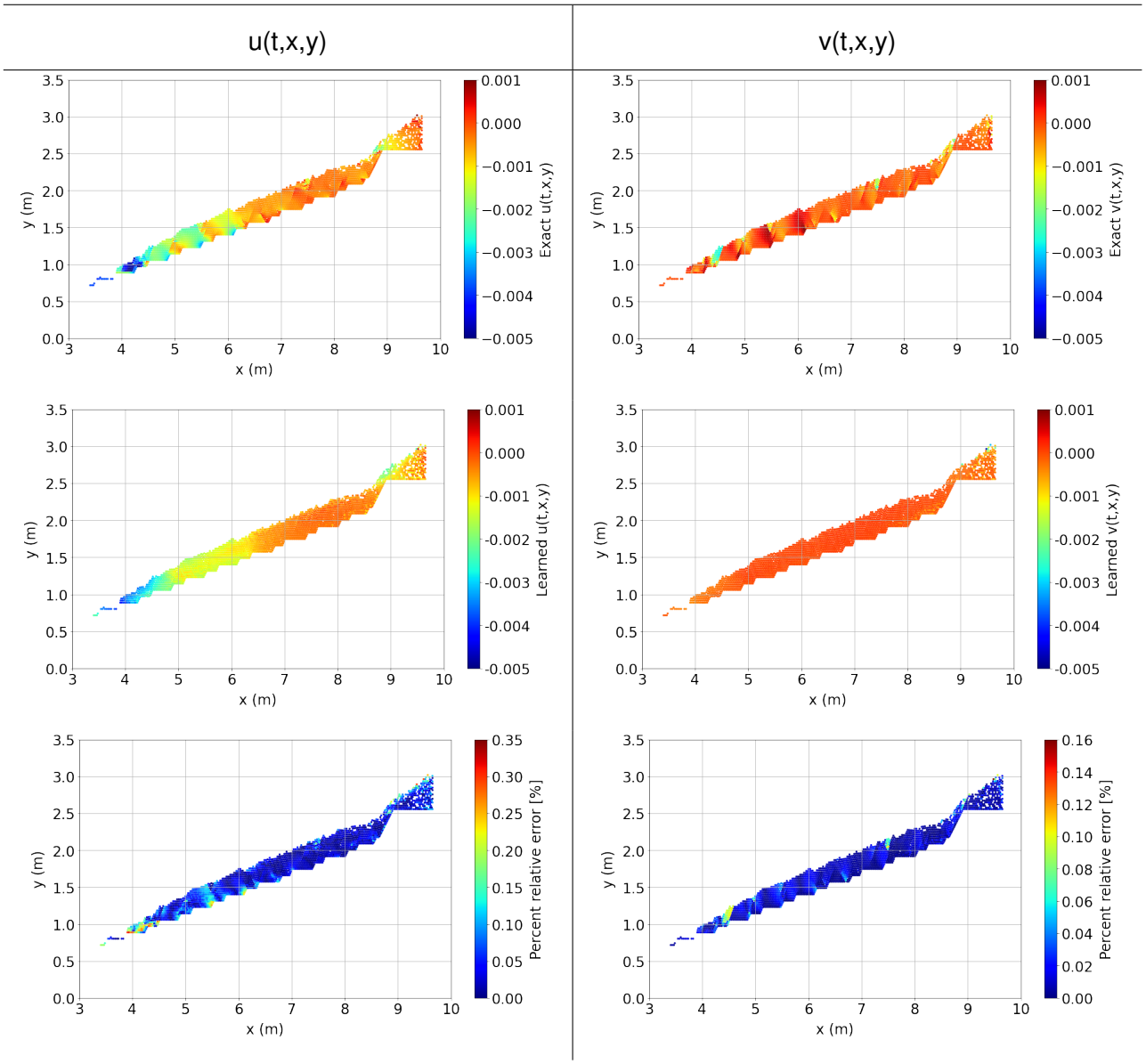


Figure 2.6: A snapshot of the velocity in the x and y directions at simulation time equal to 50 seconds. The first row displays the exact velocity value. The second row shows the predicted value for the velocity obtained from PINNs, and the last row shows the error between the exact value and the learned value, which will be used to demonstrate the framework's effectiveness.

moving forward, it contains a fewer amount of particles, and in some locations, it might not have any particles at all. That causes the results, where when we move further in time, the

maximum error value increase. Taken together, these results of predicting the velocity of the moving particles in the combustion chamber are remarkable illustrations of PINNS's advanced qualities and highlight its potential in resolving and reconstructing the physical fields.

2.5 Conclusions

In this study, the velocity of the particles moving in the bed of the biomass combustion chamber is studied by training PINNs. Traditionally, similar problems are solved by employing the DEM approach. However, in this study, we used a neural network to reconstruct the velocity fields by having sparse random data of the fields. It's worth noting that a standard neural network method does not have a prior understanding of the problem's physics and its equation of the motions. To overcome this, we used the PINN method that adds the governing equations in the loss function and used the output of the neural network to satisfy the loss function. We used the scattered data set in the spatial-temporal domain using The velocity, pressure, and density fields coming from DEM simulation. Then, we define the PINN which gets the coordinates (x and y) and the simulation time as a neural network's input and outputting two fields of velocity in the x and y direction. To avoid overfitting and under-fitting of the neural network, we tuned the neural network architecture and architecture of 10 hidden layers, and 200 neurons per hidden layer are chosen to train our neural network. The total number of the data includes approximately 2224 particles in 600-time steps. The simulation time is started at 40 seconds to avoid unstable changes at the beginning of the simulation, and it ends at 100 seconds, which gives us a total time of 60 seconds. the time step is 0.1, altogether we would have 600-time steps. The trained neural network is evaluated by comparing the learned data predicted from the neural network to the exact data provided by using DEM simulations. To do so, the plot of average percent relative error, and mean squared error in time is plotted for both velocities. The average percent error doesn't exceed the value of 1 percent and barely exceeds 0.1 percent, which

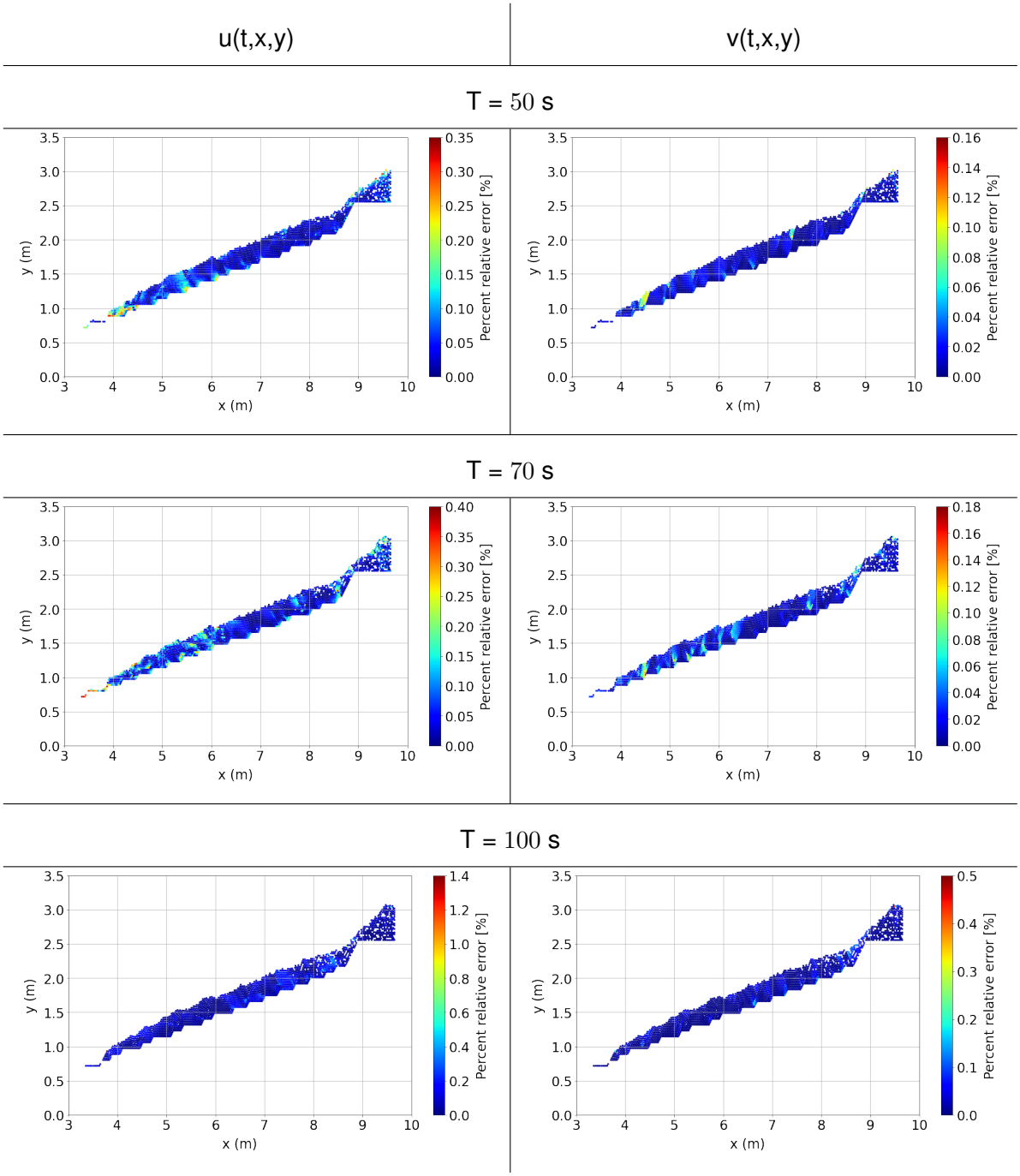


Figure 2.7: Percent error between the exact and learned velocity in the x and y direction at times 50, 70, and 100 seconds.

means that the prediction of the neural network has a good agreement with the actual data. The same trend is observed in the mean squared error which validates the reliability of the PINN in predicting and reconstructing the velocity fields.

Chapter 3

Reconstruct the biomass particles fields in the particle-fluid problem using continuum methods by applying the physics-informed neural network[1]

[1] Fateme Darlik and Bernhard Peters. “Reconstruct the biomass particles fields in the particle-fluid problem using continuum methods by applying the physics-informed neural network”. In: *Results in Engineering* 17 (2023), p. 100917.

Abstract

The motion of particles in the moving grate combustion chamber is used as the case study. These problems are categorized as particle-fluid problems. They are typically solved using Lagrangian-Eulerian methods, one of which is the coupling between the discrete element method (DEM, which is applied to the particles phase) and the computational fluid dynamics method (CFD, which is applied to the fluid phase). The current study's objective is to avoid coupling and instead, focus on using the CFD method only. There are dense piles of particles moving on the grates in the biomass combustion chamber. We assumed the dense particles' behaviors similar to the fluid, and then, applied the fluid governing equations to the particles phase. The virtual fields of the velocities, pressure, and density are specified for the particles' phase. Afterward, the physics-informed neural network (PINN) is used to reconstruct particles' fields and additionally to investigate the capability of the predicted fields to satisfy the fluid governing equations. This model has the benefit of reconstructing the particles' fields without the need for boundaries and initial conditions. The precision of the model is assessed by comparing the test data set with the exact data obtained from the eXtended discrete element method (XDEM is an in-house software). It is demonstrated that the trained neural network delivered high accuracy and is capable of predicting all outputs with an error value of less than 2 percent. Additionally, to choose the optimum architecture for the neural network, the effect of the number of hidden layers and neurons is studied.

Physics-informed neural network, Discrete element method, Continuum approach, Biomass source, particle-fluid problem

3.1 Introduction

Biomass energy is a viable source of energy and has gained attention because of its availability, accessibility, and capacity for regenerating in a short time frame. Additionally, using biomass energy considerably reduces the consumption of fossil fuels, which contributes to decreasing global warming [43]. In the power generating industry, biomass sources are

exposed to hot air and are burnt inside the combustion chamber to generate energy. One of the effective forms of the combustion chamber that is widely used in the power generating industry is the moving grate combustion chamber [44]. It provides significant benefits such as speeding up the heat and mass transport, effective temperature regulation and mixing abilities, all of which contribute to a full ignition inside the combustion chamber [45, 46]. Additionally, the efficiency of the process is affected by different parameters related to both fuel and biomass sources, such as fuel type, the moisture in the fuel, the shape and size of the biomass particles, and the airflow [47].

The biomass particle's property and their interaction over their lifetime is an important factor in simulating the biomass combustion case [48], To simulate, design, and understand the combustion of biomass sources in the power generating applications, two different methodologies can be used: Eulerian–Eulerian approach and Eulerian-Lagrangian approach [49].

In the Eulerian–Eulerian approach the biomass particles and fluids phases are both considered continuous phases. This methodology uses a simplification assumption for particle phases, which is incapable of providing detailed information about each particle. It does not simulate the discrete nature of the particle phase, and it has difficulty modeling particles' interactions and distributions [50].

The Eulerian-Lagrangian approaches are considered to be more effective. However, the Eulerian-Lagrangian approach considers the fluid phase as a continuous phase and the biomass particles as a discrete phase, which effectively addresses the above-mentioned limitations [51].

The CFD-DEM method is one of the Eulerian-Lagrangian methods. The coupling between CFD and DEM will solve particle-fluid problems in this method. The discrete phase (biomass particles) is solved using the DEM, and the continuous phase (fuels) is solved using CFD methods.

The suitability of the CFD-DEM approach in terms of predicting temperature distribution and species generation is demonstrated in [52]. Using CFD-DEM coupling, a detailed

numerical simulation of wooden biomass particles in a fixed bed is investigated [53]. The results for temperature distribution, mass fraction, and flame structures are compared to the experimental results, and good agreement is achieved.

However, CFD-DEM methods with their simplifications are still computationally expensive [54] and are not suitable for high numbers of particles in biomass combustion and their geometry complication.

To overcome the limitation, a probable alternative approach is proposed in this study that uses the machine learning method as a tool to simulate the particle-fluid problems. Machine learning methods have been widely used in the physical world in modern application areas [55, 56], as well as in mechanical engineering problems [57, 58].

An ANN is developed for an experimental fluidized bed reactor case [59]. In this study, they predict the hydrodynamics's characteristics such as minimum fluidization velocity, and maximum pressure drop which are practical optimum design and operation conditions. Likewise, in another study [60], an ANN model is developed to predict biomass gasification. The HHV of 350 samples of biomass datasets is predicted by analyzing an ANN model [61].

However, the standard neural network models are computationally inexpensive, but they show a poor sense of physics in fluid dynamics problems, and they do not have a prior understanding of the problem's physics and governing equations. To overcome this, we used a physics-informed neural network method that employs the governing equations (both continuity and Navier-Stokes equations) in the loss function of the corresponding neural network problem. Adding the equations played a key role in the regularization mechanism [62]. The PINNs are the combination of the neural network models with physical models [63], and they are useful in solving inverse problems and coupled multi-physics problems [64]. It has a wide range of applications such as additive manufacturing [65, 66], battery management systems [67], and bio mechanics [68].

PINNs use Automatic differentiation (AD) to differentiate neural networks' loss function terms regarding their inputs and model parameters to exploit the physics behind the problem. AD is one of the most valuable methods in scientific computing. It is a set of techniques to

evaluate the derivative of a function. This method has the potential to solve a wide range of fluid dynamics problems and offer new numerical solvers methods for PDEs, and a new data-driven approach for syst

This study builds on proposing a method to implement the fluid-like behavior for the dense particles in the moving grate combustion chamber. The governing equations for the fluid phases, Navier-Stokes and continuity equations, required the fields of velocities, pressure, and density to be defined. In order to model the fluid-like behavior of the dense particles, the same fields should be virtually defined. This is accomplished by calculating the number of available particles in each CFD meshed region and the average of the variables is considered as the fluids fields value. Continuing that, we used PINN to fit and map the governing equations to the particle fields (velocity, density, and pressure) in each CFD cell. PINNs follow the physical principles that drive the given data by nonlinear PDEs.

The PINN used datasets scattered in space and time as the inputs and it reconstruct the velocities, pressure, and density fields. It is important to note that the particle fields can be reconstructed without prior knowledge of the boundary and initial conditions. All in all, this study set out with the aim of assessing the PINN to model the motion of dense particles by using the equations of the motions (Navier-Stocks equations and continuity equation). Using the aforementioned strategy, instead of coupling between CFD and DEM methods to solve the particle-fluid problem, purely CFD methods might indeed be applied.

3.2 Methodology

The CFD–DEM's governing equations are built on an Eulerian-Lagrangian multiphase method. By observing the transfer of mass, momentum, and energy between the continuous and discrete phases, their equations are expanded to include the interaction between the two phases as the source term. In this method, the transport equations are used for the continuous fluid phase and the discrete phase is modeled using a discrete element method Lagrangian approach. The discrete element method treats each particle as a separate entity. The mandatory step of this method is the calculation of the force interaction. Each particle with a realistic shape and size interacts with other particles as well as the geometry of the simulation. The particles in the discrete element methods can move, rotate, and interact with one another [69].

After the calculation of particles' force interaction, the second Newton's law of dynamics is applied to each particle to compute the derivatives involved in the equations, afterward, the particle's velocity and position are updated respectively [70]. The discrete element method is however limited by truncation and round-off errors resulting from limited precision computations and the step size of finite difference methods.

In this article, the combustion chamber of the Enel Green Power "Cornia 2" power plant is considered as the case study [71]. The grates design of the combustion case split the domain into three different moving sections, as given in figure 3.1, to ensure proper mixing of the biomass particles. The grate has continuous longitudinal motion, with approximately $80\ \%$ moving forward and the $20\ \%$ being returned to the initial position, and the movement of each section switching every 30 second.

For the convenience of simulation, only the dynamic of the discrete phase is studied, and the problem's conversion potential is ignored in this stage. Accordingly, the XDEM (eXtended Discrete Element Method) method [72] developed at the University of Luxembourg is used in a high-performance computing environment to solve the biomass particles simulation and generate training and validation data for the neural network. XDEM is a novel and innovative numerical simulation technique that extends the dynamic of particles as described through

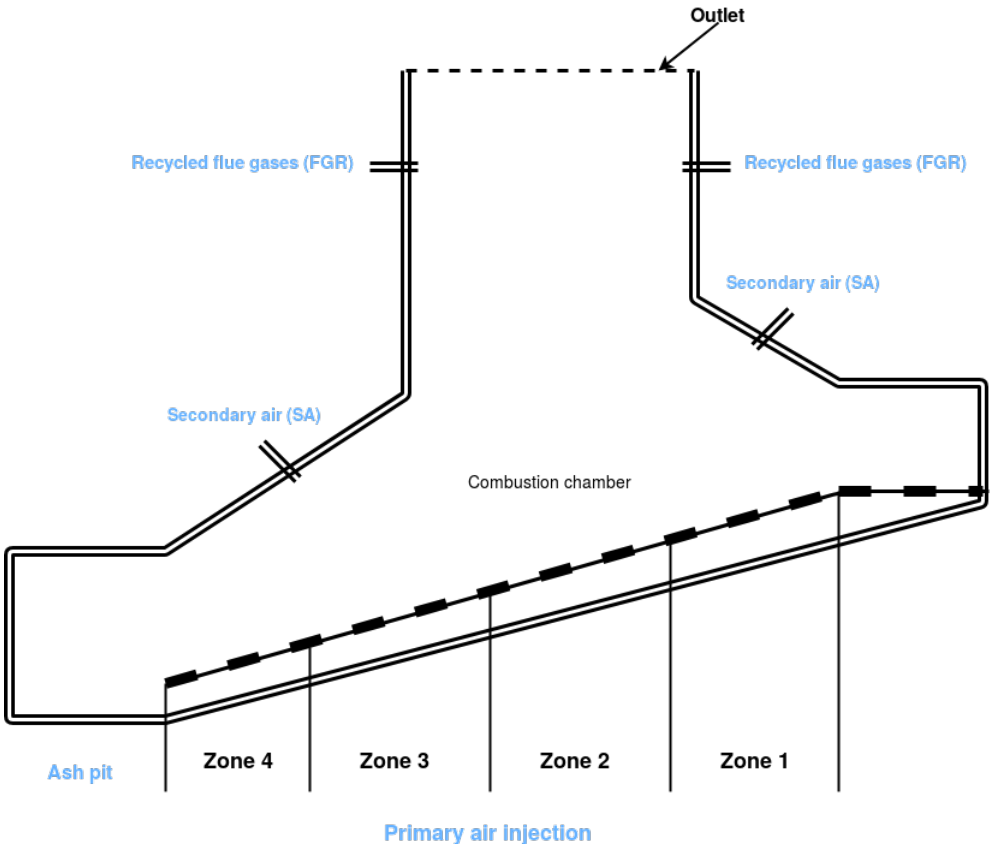


Figure 3.1: Biomass furnace design [71].

the classical discrete element method by additional properties such as the thermodynamic state for each particle [73].

Generally, the particle bed is composed of a mixture of approximately $79.3\ \%$ wood chips and $20.7\ \%$ agricultural wastes, which resembled the real-world conditions of the biomass plant. In the present study, the biomass bed distribution consists of 1761 number of beach-wood particles and 461 number of stringy-wood particles, and all the particles are assumed to have the same volume with a radius of $0.015\ cm$ (figure 3.2). The particles are entering the domain from the right side, and from the left, they will leave the domain. Additionally, a mass flow with the rate of $0.0155\ kg/s$ for beach-wood particles and $0.00388\ kg/s$ for stringy-wood particles are entering the domain from the input section. The properties of particles are reported in the table 3.1.

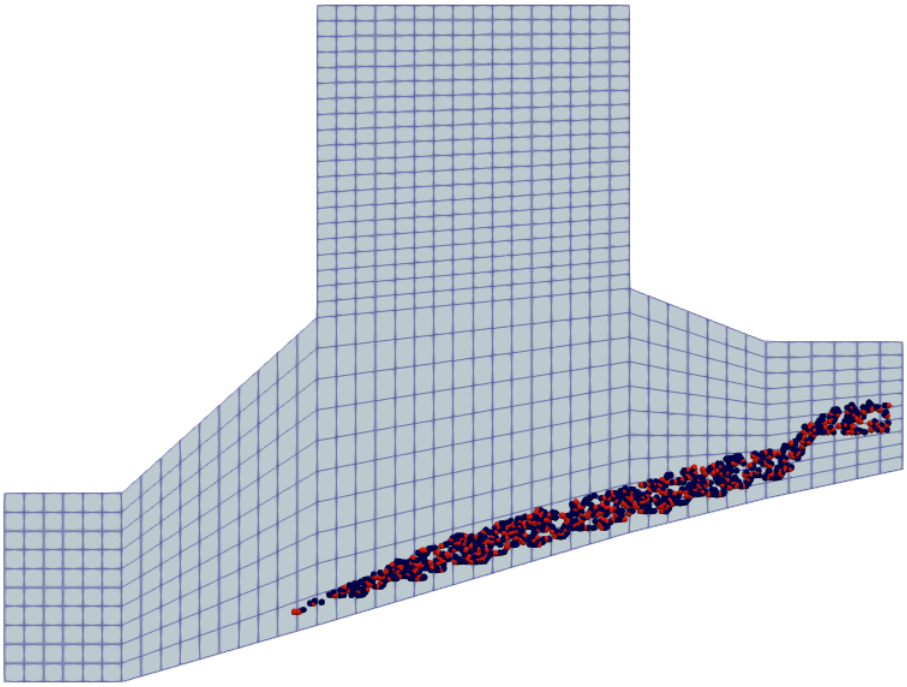


Figure 3.2: Blue particles are beach-wood particles and the red ones are stringy-wood particles.

As it has been explained, this study used a physics-informed neural network to model the behavior of dense particles on the moving grate by applying the equations of the motions (Navier Stocks equations and continuity equation). Using the aforementioned strategy, one can replace the coupling of CFD and DEM methods to simulate particle-fluid problems by purely using the CFD method which will be useful in terms of time-consuming and overcoming CFD-DEM limitations.

The computational fluid dynamics method is a branch of science that solves fluid mechanics problems. Generally, CFD methods require meshed geometry, the boundary and initial conditions (depending on the dimensions and the governing equations of the problem), and identified physical modeling. In the next step, the governing equations are

Table 3.1: Particle Characteristics of Beach-wood and Stringy-wood Biomass

Properties	Beach-wood	Stringy-wood
Particle radius [m]	0.015	0.015
Mix fraction [$\%$]	79.5	20.5
Mass flow [kg/s]	0.0155	0.0038
Number of particles [$-$]	1762	462
Apparent density [kg/m^3]	648.52	610.47
Mean porosity [$-$]	0.685196	0.6375

solved iteratively, and the convergence and stability of the CFD methods are checked.

In the biomass simulation, the geometry of the combustion chamber is taken into account. Accordingly, the position of the biomass particles which are distributed into the defined domain is taken and based on that, the number of particles available within each discrete cell is calculated. It should be noted that, if one particle is located amongst two or more cells, the proportions of its presence in each cell are appended to the total number of the particles in that cell. Furthermore, we calculated the mean density and velocities of the particles by using the following equations (3.3) to (3.2).

$$\overline{u_j} = \frac{1}{N_j} \sum_{i=1}^{N_j} u_{ji} \tag{3.1}$$

$$\overline{v_j} = \frac{1}{N_j} \sum_{i=1}^{N_j} v_{ji} \tag{3.2}$$

$$\overline{\rho_j} = \frac{1}{N_j} \sum_{i=1}^{N_j} \frac{m_{ji}}{V_j} \tag{3.3}$$

Where in cell number j, V_j is the cell volume, N_j is the total number of particles

available in the cell j, $\overline{\rho_j}$ is the density, $\overline{u_j}$ is the horizontal velocity, and $\overline{v_j}$ is vertical velocity. Additionally, for particle number i present in cell number j, m_{ji} is the mass, u_{ji} is the velocity in the x-direction, and v_{ji} is the velocity in the y-direction. To calculate the pressure the approach suggested by Qi et al [74] is used. First, the total stress tensor in the cell number j called σ_j is calculated as follows:

$$\sigma_j = \frac{1}{N_j V_j} \sum_{i=1}^{N_j} \sum_{x=1}^{X_i} -F_{ix} \otimes l_{ix}. \tag{3.4}$$

Where for particle number i, X_i is the number of particles-particles interactions, F_{ix} is the interaction force, and l_{ix} is the vector connecting the centers of particles' existence in the interactions. The terms in equation (3.4) indicate the dynamic and static components, correspondingly. The stress σ_j is broken down into two parts, an isotropic portion P_j and a deviatoric part τ_j, in which the P_j is stated as:

$$P_j = -\frac{1}{3} tr(\sigma_j) \tag{3.5}$$

The total pressure in each cell is calculated using the equation (3.5). Afterward, the neural network is used to complete the mission. This study is motivated by modern methods for solving forward and inverse problems including compressible Navier Stokes equations 3.6 to 3.8 where the solution is approximated by a neural network. Furthermore, the prioritization of the solution is fully justified by a similar approach that extends unknown solutions in terms of a proper set of basis functions. To tackle the problem, we started by estimating the implicit function using a proposed neural network that outputs four variables $u(t, x, y)$, $v(t, x, y)$, $\rho(t, x, y)$, and $P(t, x, y)$ and takes as input t, x, and y.

$$\frac{\delta \rho}{\delta t} + \frac{\delta(\rho u)}{\delta x} + \frac{\delta(\rho v)}{\delta y} = 0 \tag{3.6}$$

$$\frac{\delta(\rho u)}{\delta t} + \frac{\delta(\rho u^2)}{\delta x} + \frac{\delta(\rho uv)}{\delta y} + \frac{\delta P}{\delta x} - \mu(\frac{\delta^2 u}{\delta x^2} + \frac{\delta^2 u}{\delta y^2}) = 0 \tag{3.7}$$

$$\frac{\delta(\rho v)}{\delta t} + \frac{\delta(\rho uv)}{\delta x} + \frac{\delta(\rho v^2)}{\delta y} + \frac{\delta P}{\delta y} - \mu(\frac{\delta^2 v}{\delta x^2} + \frac{\delta^2 v}{\delta y^2}) = 0 \tag{3.8}$$

3.3 Neural network model

A physics-informed neural network is a powerful tool for solving complicated problems [75], nonlinear systems involving multi-dimensional input data [76]. Based on the recent improvements in physics-informed neural network [77, 78, 79], it is capable of estimating unknown fields to predict the partial differential equation solution.

To predict the solution of variable u and the nonlinear function N, the physics-informed neural network uses the simple neural network in conjunction with the physical model f specified in equation (3.9).

$$f := u_t - N(t, x, u, u_x, u_x x, ..) \tag{3.9}$$

The derivative of the neural network variable u concerning time t and space x is calculated using automated differentiation which is among the most important tools in scientific computing. Automatic differentiation uses the chain rule of derivative components [80].

Numerical differentiation is the process of determining the numerical value of a derivative, and it needs the finite difference approximation of derivatives using values of the function evaluated at certain points. Due to the restricted computational precision and the selection of the finite difference step size, which both causes truncation and round-off error, numerical approximations of derivatives are naturally imprecise and unstable. Symbolic differentiation, on the other hand, modifies expression in algebra systems to calculate the derivative of a formula with a specified variable and creates a new one as its result. Symbolic derivatives generate new formulas by modifying existing ones rather than performing numerical calculations which necessitate a closed-form model that either excludes or significantly limits the rate of algorithmic control. Without a proper introduction, automatic differentiation is misunderstood as numerical or symbolic differentiation. Automatic differentiation is based on

the idea that all numerical calculations ultimately comprise a limited set of basic operations where the derivatives are defined.

The derivative of the whole composition is obtained by combining constitutive operations achieved by the chain rule. Generally, automatic differentiation, in contrast to derivative expressions, depicts numerical values of derivatives using symbolic differentiation procedures (but not the resulting expressions), which is misleading. The automatic differentiation can be described as partly symbolic and partly numerical [80].

TensorFlow, a well-known and well-documented open-source software package are beneficial for deep learning operations. TensorFlow also provides an automatic differentiation tool, making it a helpful tool for solving PINN problems. Additionally, it also builds the computational graph explicitly before running the model. The above permits you to design and compile a computational graph of a deeply hidden physics model at just one time, and preserve it in that state all through the training step. The computational complexity of the suggested methodology will be greatly reduced as a result of this.

One of the most important aspects of any neural network is the choice of the loss function. In general, the loss function is used to guarantee that the magnitude of the error between the prediction and the exact output data is within acceptable bounds. The error between the prediction and the exact output data decreases when the value of the loss function decreased. In this study, the mean squared error, which takes the mean of the squares of the errors, is chosen as the loss function.

Regression loss is only one component of the PINN's loss function. It also includes the differential fluid governing equations. The neural network loss function should guarantee that the predicted variables are capable of satisfying the equations in addition to guaranteeing that they are close to the exact variables. 3.10.

$$MSE := \sum_{i=1}^{N} \left(|u(t^i, x^i) - \hat{u}(t^i, x^i)|^2 + |f(t^i, x^i)|^2 \right) \tag{3.10}$$

Where $u(t^i, x^i)$ is the training datasets, and $\hat{u}^i$ is the exact data at the time t^i and in the

position x^i. The term $|u(t^i, x^i) - \hat{u}(t^i, x^i)|^2$ aims to fit the data by adjusting the neural network parameters, and the term $|f(t^i, x^i)|^2$ efforts to solve the partial differential equations.

The learned equations can indeed be solved, and the outputs can be evaluated to the exact solution of partial differential equations. In figure 3.3, the physics-informed neural network architecture is demonstrated. The neural network's inputs are time t and coordinates (x and y), and the outputs are four fields composed of two velocities, pressure and density, and fluid's PDEs equations are added to the loss functions.

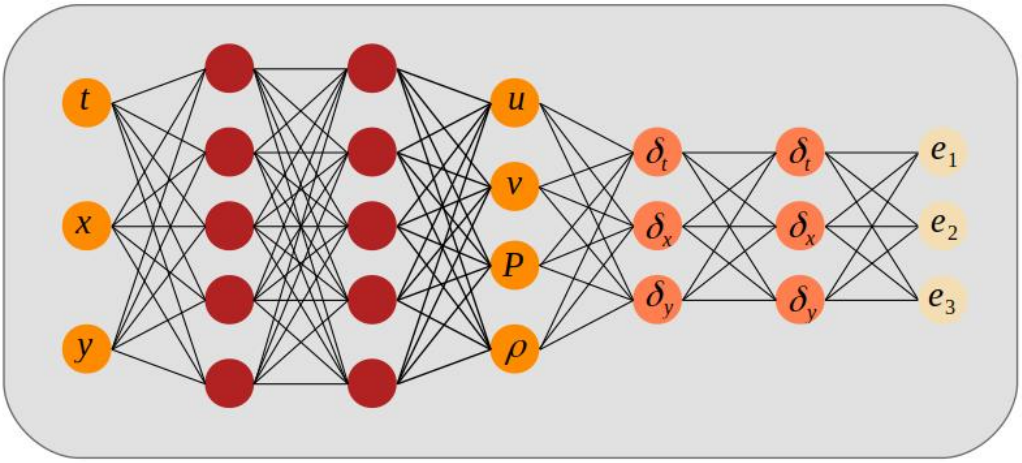


Figure 3.3: Physics-informed neural network architecture.

To have a general overview, firstly, the data set for all four fields of two velocities, pressure, and density are collected from XDEM simulation. Afterward, the proposed physics-informed neural network along with PDEs equations (Navier Stokes and continuity equations 3.6, 3.7, and 3.8) is used to train the neural network. Following that, the fields of the particle-based simulation are reconstructed using the trained neural network.

The automatic differentiation is used to obtain the required derivatives to compute the residual networks from equations (3.6) to (3.8), which are named e_1 to e_3, respectively. The shared variables of the neural networks u, v, P, and ρ as well as the equations e_1, e_2, and e_3 are learned by minimizing the mean squared error loss function mentioned in equation (3.11)).

$$
\begin{aligned}
MSE := \sum_{n=1}^{N} \Big(|u(t^n, x^n, y^n) - \hat{u}^n|^2 + \\
|v(t^v, x^n, y^n) - \hat{v}^n|^2 + \\
|P(t^n, x^n, y^n) - \hat{P}^n|^2 + \\
|\rho(t^n, x^n, y^n) - \hat{\rho}^n|^2 \Big) + \\
\sum_{i=1}^{3} \sum_{n=1}^{N} (|e_t(t^n, x^n, y^n)|^2)
\end{aligned}
\tag{3.11}
$$

Where N is the number of training points, the first four terms account for the two velocities, pressure, and density data set, and the last term defines the effects of Navier Stokes and continuity equations on the loss function. As mentioned earlier, training data set points are distributed in space and time. The entire 266640 data set produced by the 2222 particles in 120 times snapshots spread in space and time are used for both field regression and the accompanying partial differential equations.

The activation function is mandatory to process the neuron node's input value and produce the output value. The activation function is used to modify the data set's initial linear connection and add the nonlinear transition into the neural network to improve the fitting capacity.

The activation function can significantly improve the neural network's capability to control, evaluate, and anticipate complicated nonlinear problems. In the current research, the activation function $relu(x) = max(0, x)$ is used for the hidden layers in the physics-informed neural network.

The network displayed in figure 3.3 for illustration reasons consists of two hidden layers with six neurons in each. In the mentioned figure, the differential operator $\delta_t, \delta_x, \delta_y$ is computed using automatic differentiation.

The optimization process of the neural network is mainly realized through gradient descent. The Adam optimization method is a stochastic gradient descent approach that is

highly efficient, requires low memory, and is appropriate for large data problems [42].

In ADAM's optimization process, the learning rate aims to establish a perfect compromise between performance and speed. The learning rate is a configurable optimization method's hyper parameter that controls the step length at each iteration while progressing toward a loss function minimum.

The learning rate controls how quickly the neural network model reacts to changes. If the learning rate is too high, fewer epochs are required which leads to fast identification of neural network parameters. However, this might result in instability of the training process and failure to converge. However, if the learning rate is too low, even though the accuracy is acceptable and the neural network is capable of correctly predicting the data set, the training will be substantially slower due to the small changes applied to the weights within each epoch. Therefore, the adaptive learning rate strategy is used to adjust the learning rate value within the training procedure. Furthermore, using an Adam optimizer with an adaptive learning rate, the neural network parameters are trained to rely on the gradients of the loss function.

Another model's hyper parameter is the batch size. Increasing the number of batch samples can contribute to decreasing the learning rate value. Batch size is applied to the training data set, and it determines the rate of gradient decline. Using the batch size is beneficial since each batch has fewer data and decreases the cost of each epoch's processing.

The batch size should be chosen based on the speed and precision of neural network training throughout multiple batches.

To allow the network to predict all four outputs concurrently, we normalize both the input and output data to fit within the range $(0, 1)$. Given that the variables have different ranges, normalization can help the model's convergence.

During training, the neural network's performance is also assessed on the validation data set to guarantee that the training loss value does not drop while the validation loss value grows. Fine-tuning the neural network hyper parameters are mandatory to find an optimal

value of the model's parameters that are large enough to capture all the information while being time-efficient. Generally speaking, having a deeper neural network (more hidden layers) or a wider neural network (more neurons) results in having a greater number of model parameters, which can increase the accuracy of the training dataset but may require an extended duration to learn the model, and might leads to overfitting. When the model began to overfit, the accuracy of the validation data set will begin to drop.

3.4 Results and discussions

The aim of this study is to consider the fluid-like behavior of dense particles in the moving grate combustion chamber. The particles are entering the combustion chamber, and after exposing to the hot air, they will exit from the other side of the domain. The grate's back-and-forth motion will allow the particles to completely burn during the process. A total of 2222 wooden particles are present in the domain.

Due to the lack (difficulties) of access to experimental datasets, we used the datasets generated from the simulation. The simulation is done by using the XDEM (in-house software). After collecting the particle's datasets (positions, velocities, and interaction forces), we calculated the virtual fields required to solve the fluid governing equations, by mapping the data into the existing CFD meshed region. The governing equations on the fluid phases are the Navier-Stokes and continuity equations. Following that, the governing equations are added to the PINN's loss function. As a result, in addition to reconstructing the aforementioned fields as outputs, the PINN also verifies that the given virtual fields satisfy the fluid governing equations. The 30 percent of the total datasets are considered as the test dataset, while the remaining datasets are assigned to the train datasets, from which the 10 percent is used as a validation dataset.

Before all else, training and testing a neural network necessitates identifying the structure of the neural network and choosing the optimum number of the neural network's hyperparameters.

Building neural networks, in fact, is an intuitive approach [81]. As a general rule, the neural network with a high number of parameters is able to learn more complex models with non-linearity. However, it might lead to overfitting and a poor generalization of test data. In contrast, a neural network with a less number of parameters will run faster, but might leads to under-fitting and might have an inaccurate prediction in both train and test detests.

The accuracy of the model is measured by calculating the mean squared error between the estimated network and the target function, and by minimizing it, we seek to obtain the optimal hyperparameters. The following paragraphs present the findings from the

hyperparameter's fine-tuning. To study the effect of changing the number of hidden layers on the neural network's evaluation, the mean squared error loss function value is plotted against the number of hidden layers in figure 3.4, while all other hyperparameters remained constant. These findings back up the notion that as the number of layers increases, the training loss function value decreases. The presence of overfitting occurs when the validation loss function value increases. By way of explanation, adding additional hidden layers improves the model's accuracy up to a roughly certain point, but once that point has been reached, adding more layers could severely undermine the model's performance. The six hidden layers were selected as they had the minimum loss function value on the validation dataset. In figure 3.5, the mean squared error loss function value for a network of six hidden layers is plotted against the number of neurons. As would be expected, the value of the training loss function is dropping as the number of neurons increases.

Nevertheless, the validation loss function value first shows a decreasing trend, which then turns into an increasing trend after a specific point. By comparing the plot of the number of neurons with loss functions for both training and validation data sets, the thirty number of neurons appears to be the optimal value.

For further analysis of the neural network performance, the learning rate factor of the Adam optimization method is used to adjust the parameters in the neural network in order to minimize the mean squared error during the training of the neural network. An adaptive learning rate algorithm depends on the inverse time decay used in the current studies.

Once the model is trained, the performance of the trained neural network can be evaluated by comparing the results of the XDEM simulation with the learned outputs from the neural network.

To illustrate the effectiveness of the current approach, the results of the time mean squared error and time mean relative percent error will be discussed separately in the following paragraphs. The time-mean squared error is calculated as shown in equation (4.22).

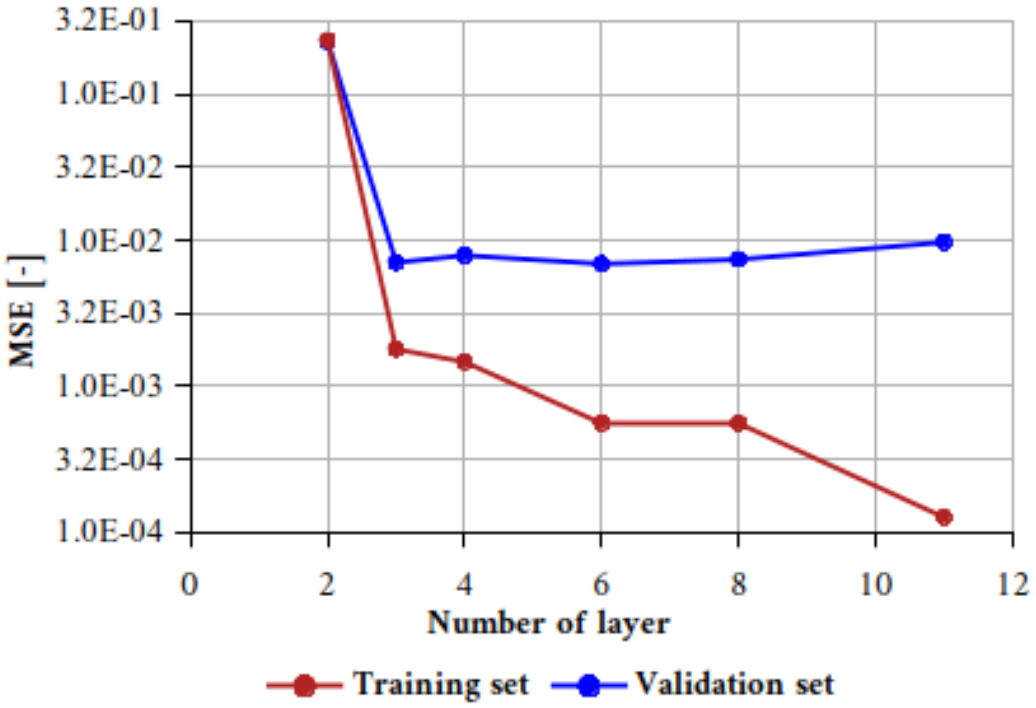


Figure 3.4: Comparison of the MSE for different numbers of layers in the physics-informed neural network model.

$$MSE^t := \frac{1}{N}\sum_{i=0}^{N}(y_i^t - \hat{y}_i^t)^2 \tag{3.12}$$

Where N is the total number of particles, y is the exact value, and $\hat{y}$ is the learned value in the time t. This metric gets the average value of the mean squared error in all the cells in the time t. The results of the time mean squared error for all four output variables (including two velocities, pressure, and density) are depicted in figure 3.6 which is quite rational. It can be seen from the figure, that the value of the time mean squared error will drop gradually along with the flow of time. It is necessary to point out that, the proposed framework is capable of reconstructing all the fields with the accuracy of order 10^{-3} to 10^{-6}.

One anticipated finding is that the velocities' time mean squared error has a greater value at the beginning of the simulation. It can be explained by the fact that the neural

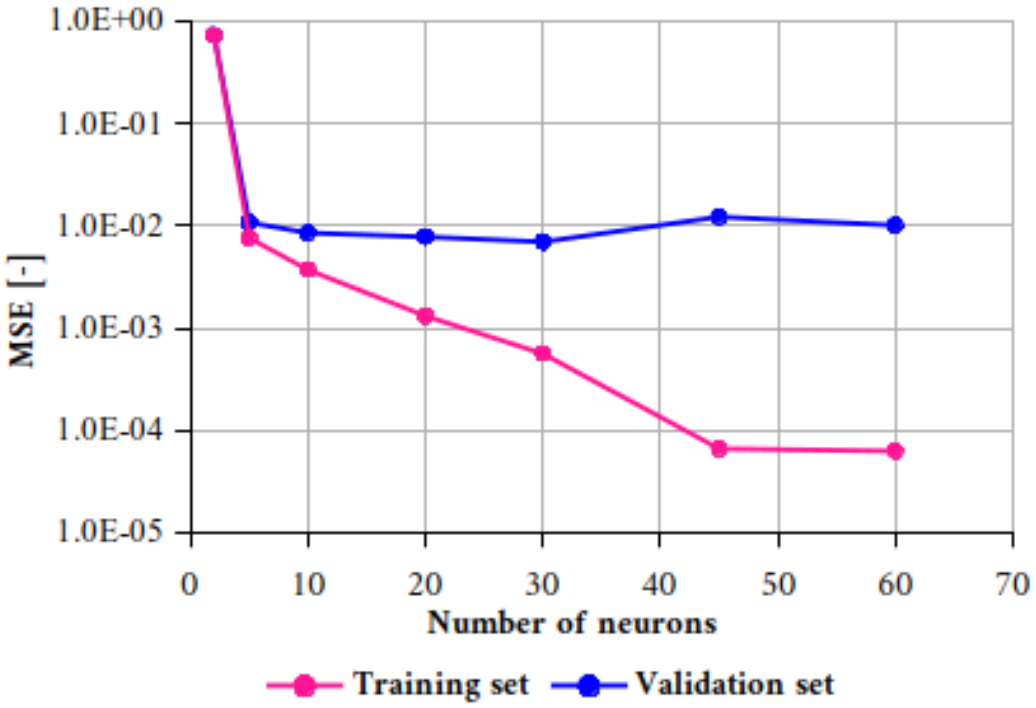


Figure 3.5: Comparison of the MSE for different numbers of neurons per hidden layer in the physics-informed neural network model.

network finds it challenging to accurately represent the physics of the problem due to a lack of sufficient data and unexpected changes at the beginning of the simulation. On the other hand, when the simulation is ongoing, the neural network can perform a more accurate prediction.

A more intriguing result comes from the mean relative percent error (MRPE) which measures the accuracy of the prediction, and is calculated as follows:

$$MRPE^t := \frac{1}{N} \sum_{i=0}^{N} \frac{y_i^t - \hat{y}_i^t}{y_i^t} \tag{3.13}$$

Where N is the total number of particles, y is the exact value, and $\hat{y}$ is the learned value in the time t. The plot of MRPE values over time is shown in figure 3.7. According to the findings from the MRPE, the maximum values of the fields almost never exceeds $1,8\%$. As

seen in the referenced figure 3.7, the density's MRPE is generally lower than that of the other fields. That can be explained by the fact that the PINN added the continuity equation to its loss function. Given that, the PINN has a better grasp of the density field, which leads to better predictions.

In contrast to the other two fields, the velocity in the y-direction and pressure had a higher error value, as seen in figure 3.6. It may be inferred that the velocity in the y-direction error value is higher because gravity exists and causes higher acceleration and sudden changes in the velocity in the y-direction. The pressure field error can be explained by the fact that there is no precise equation for pressure in the method of applying equations to the PINN loss function and only the derivatives of pressure are calculated.

In summary, the figures show that the learned outputs are generally in good agreement with the test data.

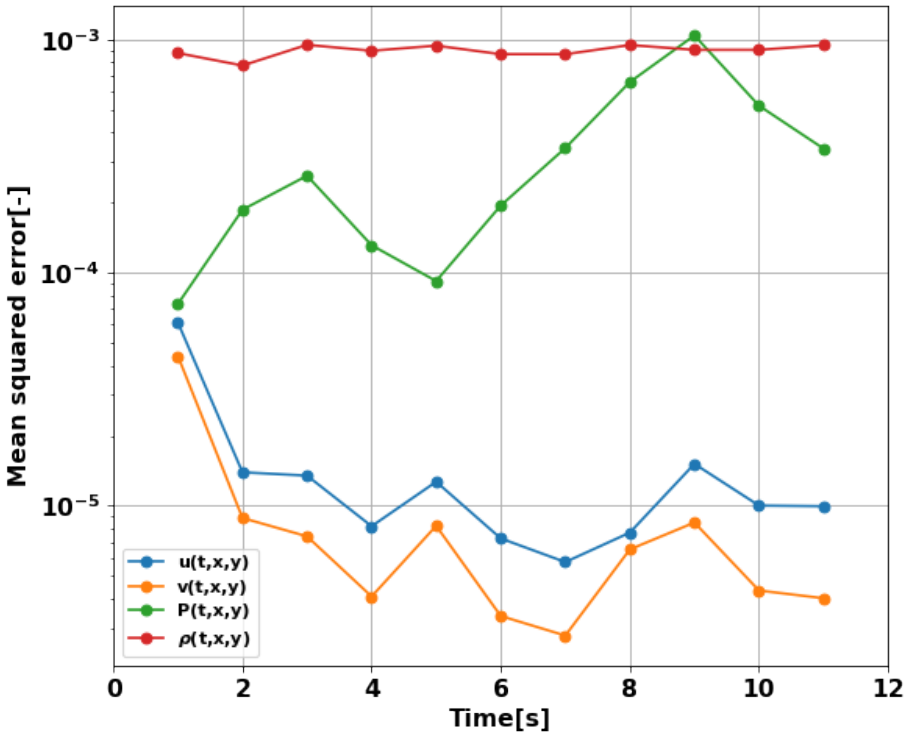


Figure 3.6: Time mean squared error of the model predictions and the corresponding exact velocities, pressure, and density fields

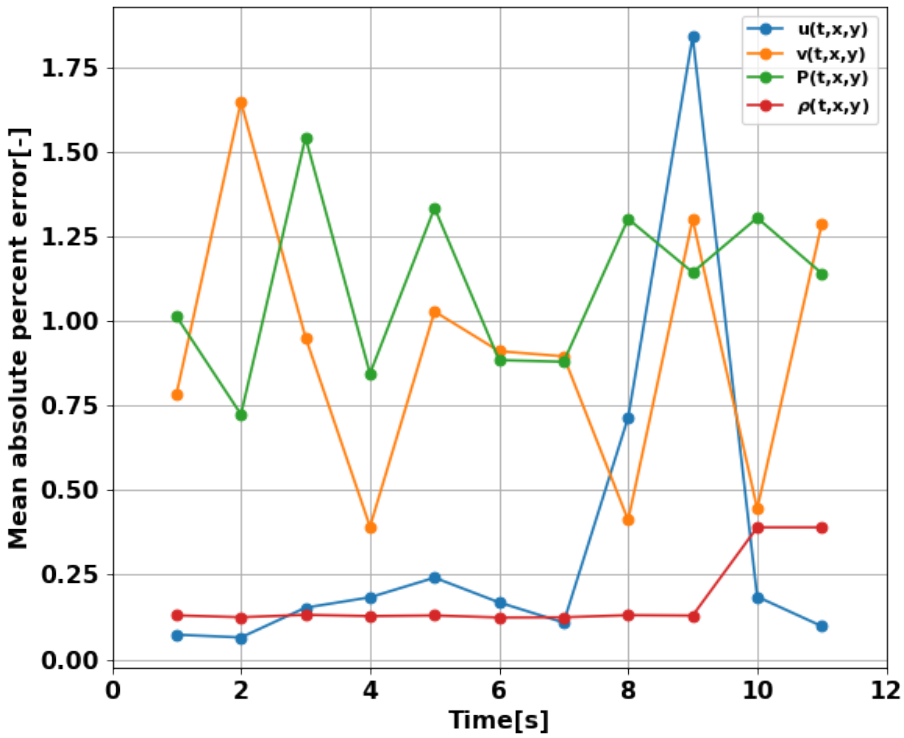


Figure 3.7: Mean relative percent error of the model predictions and the corresponding exact velocities, pressure, and density fields during the time.

figures 3.8 to 3.11. The predicted fields of the neural network are compared graphically with the simulated fields at a time of 10 seconds using the trained neural network (figures 3.8 to 3.11). The trained neural network shows excellent robustness for prediction and reconstruction across all fields, as it is demonstrated. Each of the fields is covered in detail in the paragraphs that follow.

A visual comparison of the exact velocity field is presented in figures 3.8 and 3.9. Looking at velocity components in figures 3.8 and 3.9, As can be observed, the ranges of the velocity's percent relative error in the x-direction and y-direction are 1% to 5.6% and 2% to 7%, respectively. In general, the error of velocity in the y direction is higher than the error in the x direction. It demonstrates that the neural network has difficulties learning the values of the vertical velocity rather than the horizontal one. We attribute this to the fact that the gravity in the y-direction added extra acceleration to the particles' motion. Particularly,

the acceleration rises significantly, when particles move between the grates (falling from their current grate to the next neighbor grate). The trained neural network's ability to predict the vertical velocities in these circumstances appears to be impacted by an unanticipated increase in particle acceleration.

Looking at figures 3.8 and 3.9 demonstrates that the majority of the percent error happened at the beginning part of the domain, where the mass flow enters the domain. This also makes sense, because compared to the other parts of the domain, the beginning part is where we face sudden changes in the number of particles. Every 1 second, fresh biomass particles are entering the domain and this directly affects the density of cells. Furthermore, the velocities of the dropping particles will significantly affect the velocities.

Figure 3.10 compares the learned pressure obtained from the neural network to the actual pressure data sets and shows a good agreement between predicted and exact pressure values. The proposed framework is capable of accurately predicting the pressure with a relative percent error of $0.6\ \%$. Figure 3.10 demonstrates that the pressure's relative percent error has one order of magnitude higher value than other fields. This finding may be explained by the fact that the pressure fields are not defined directly in the fluid governing equations, and only the pressure derivatives are used. Furthermore, it can be explained by the method used to calculate the fluid phase's pressure in the first place, which involves a significant set of assumptions. A representative snapshot of the predicted density field is illustrated in figure 3.11, which is shown to be in good agreement with the exact data set.

The potential of the PINN to precisely reconstruct the density fields is proven by the addition of the Navier-Stokes compressible equations and continuity equation to the PINN loss function.

The results highlight the capability of the trained neural network to predict the density field with an average relative percent error of 1.5% and a maximum relative percent error of 3.5%.

The maximum density error value occurs primarily at the domain's boundary (input, output, and near the grates). The assumption of having dense particles can be used to

explain it. If there is insufficient particle concentration, this assumption loses its validity. The neural network, which is based on the physics of the problem, is unable to predict the fields as accurately as the other domain with dense particle piles.

Taken together, the results of predicting fluid flow fields are remarkable illustrations of PINN's advanced qualities.

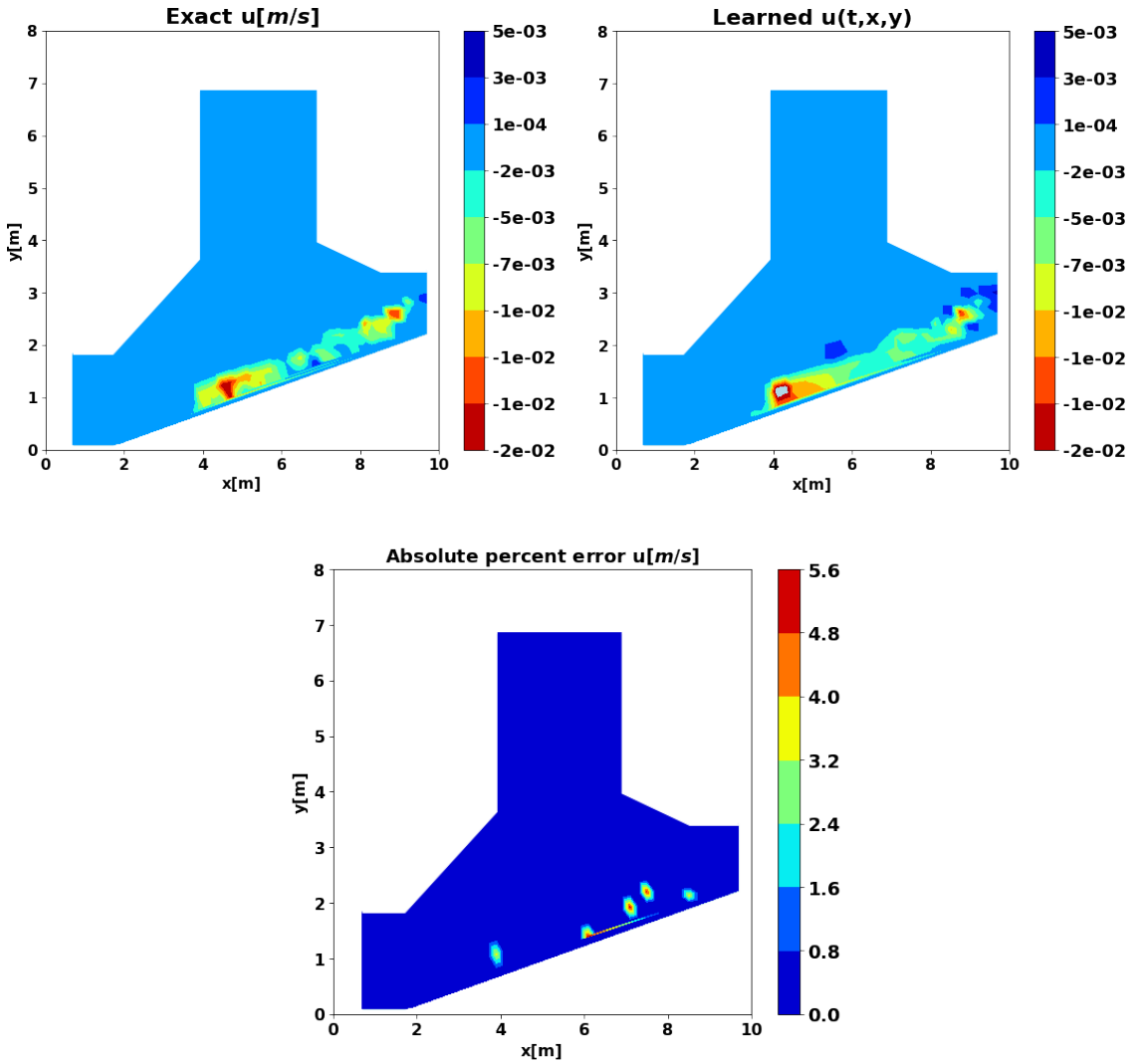


Figure 3.8: A representative snapshot of the normalized velocity field in the x-direction at t=10 seconds

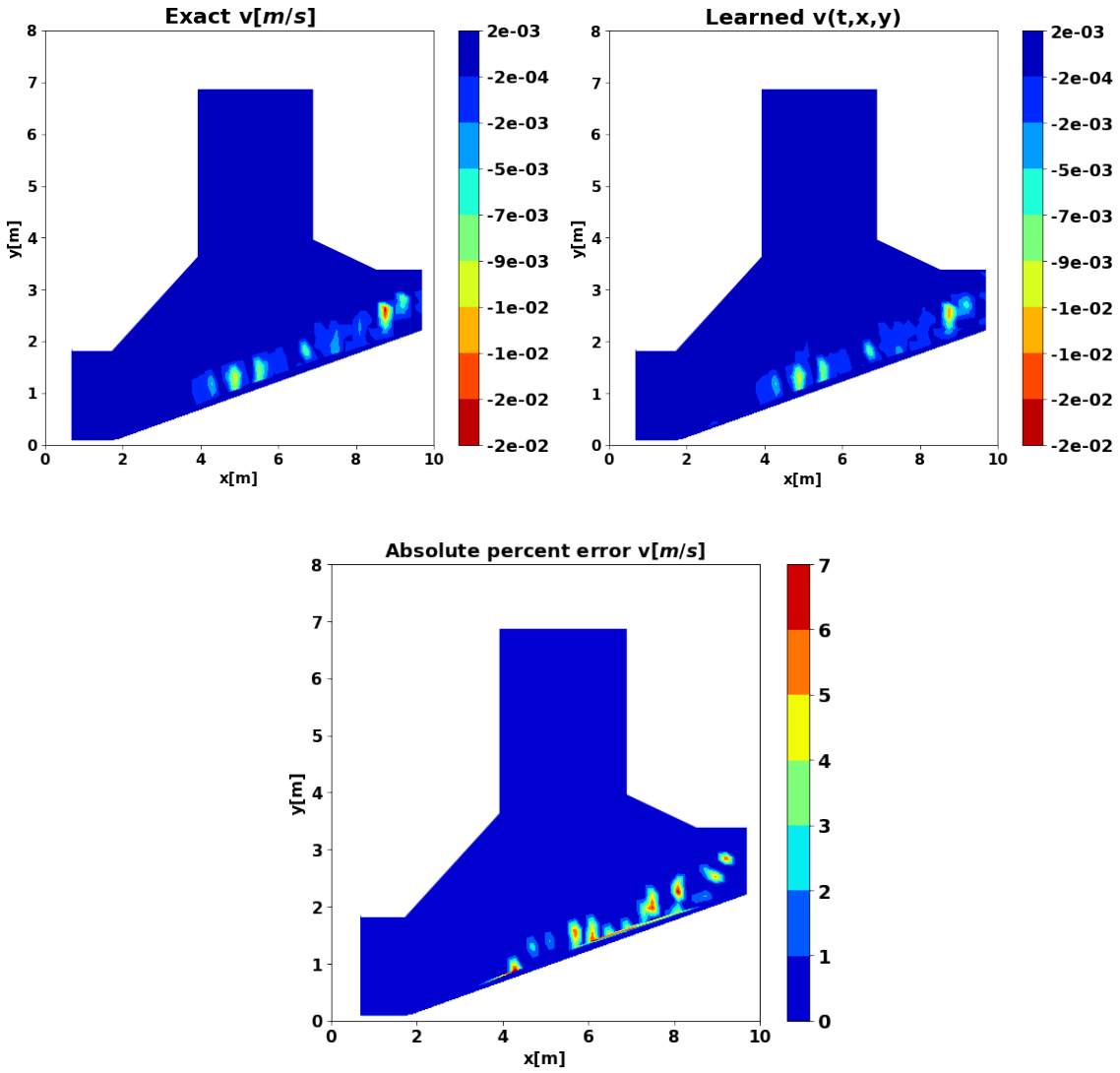


Figure 3.9: A representative snapshot of the normalized velocity field in the y direction at t=10 seconds

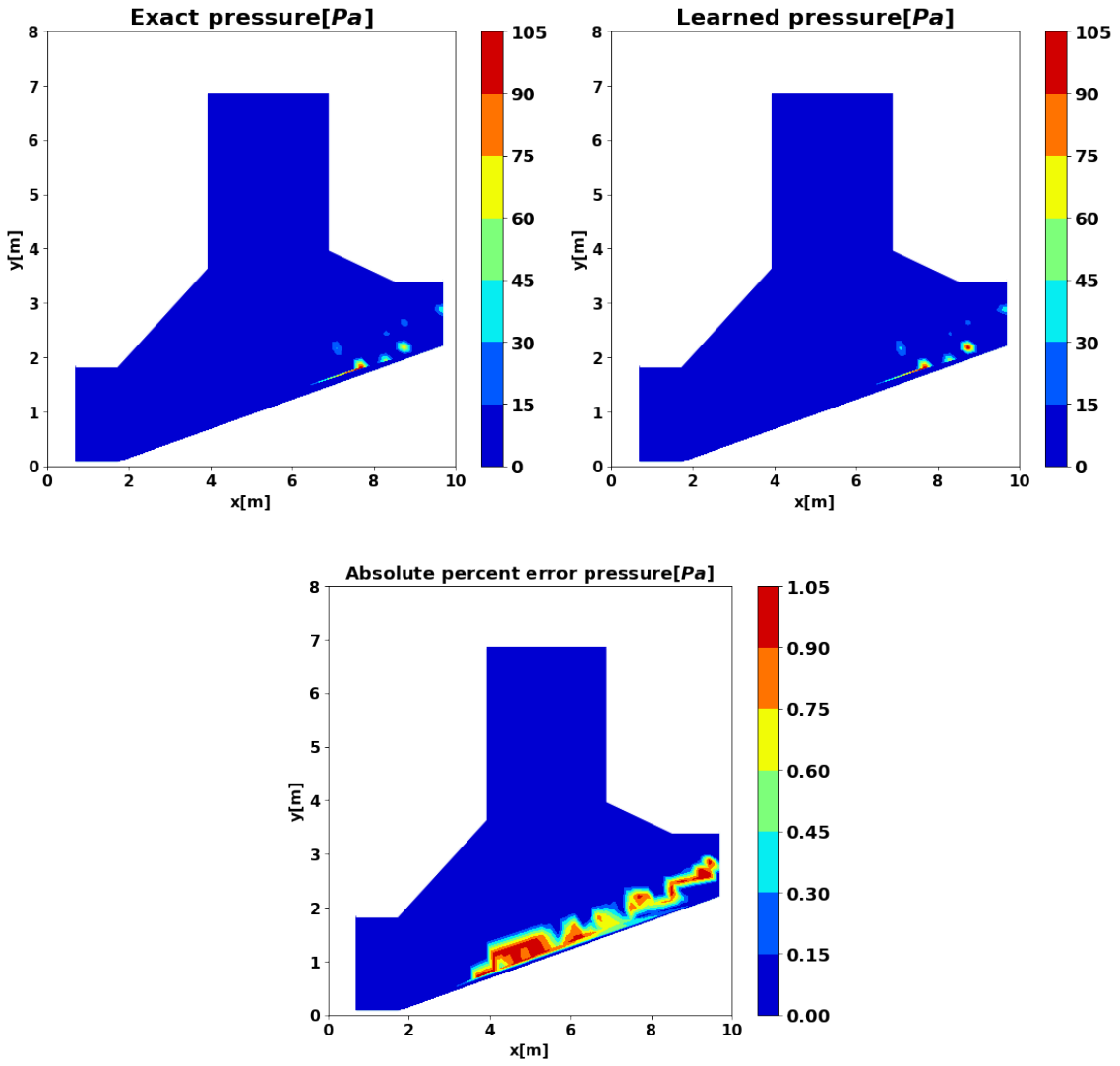


Figure 3.10: A representative snapshot of the normalized pressure field at t=10 seconds

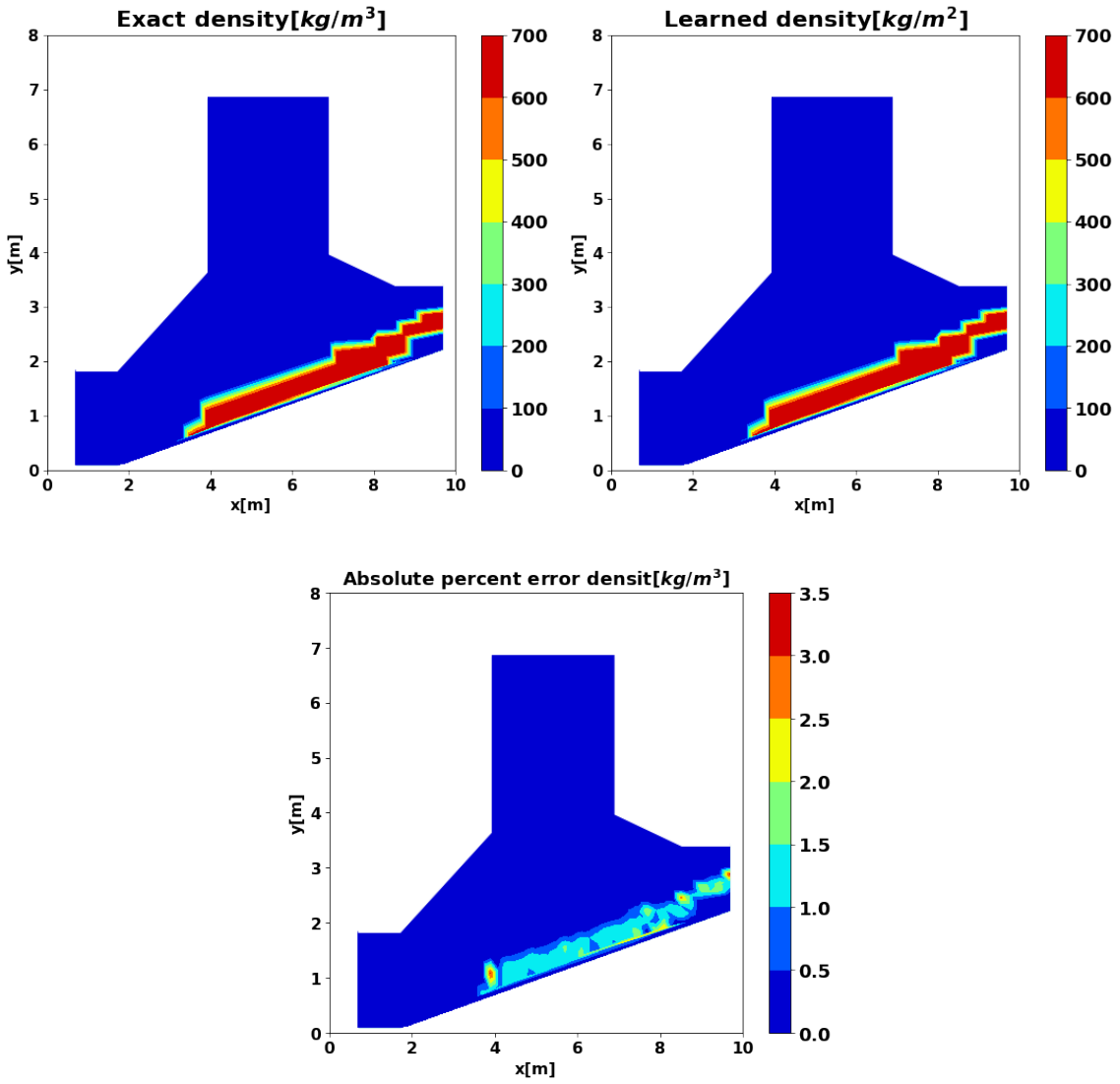


Figure 3.11: A representative snapshot of the normalized density field at t=10 seconds

3.5 Conclusion

In the present paper, the motion of the biomass particles in the moving grate combustion chamber is studied. This problem belongs to the category of particle-fluid problems, which are often solved by Eulerian-Lagrangian methods. One common approach is the coupling of the DEM (which focuses on solving the discrete part) and the CFD methods (which focuses on solving the continuous part of the simulation). Given the number of biomass particles entering the combustion chamber, we assume to have dense piles of particles. The current study aims to assume the behavior of dense particles similar to the fluid. The four variables required to solve the fluid problem are velocities, pressure, and density. Accordingly, we should establish similar virtual fields for the particle phase as well, and they should be able to satisfy the fluid governing equations. In this study, two piles of particles with the same volume and distinct compositions (Beach-wood and Stringy-wood) can be observed in the combustion chamber. The domain has a total of 2222 numbers of particles available.

We extract the virtual fields from the scattered particles datasets and map them into CFD meshed region, which has been comprehensively described in the section 4.4. The dataset required for the training of our neural network is obtained from the in-house software XDEM.

After defining the essential fields, the PINN is used to reconstruct the particles' fields. The PINN gets the coordinates and time (x, y, and t) as its input and it predicted the four mentioned fields (u, v, P, and ρ) as its outputs. Furthermore, the fluids' governing equations (Navier Stokes equations and continuity equation) are included in the loss function, forcing the predicted variables to satisfy the equations. First and foremost, to gain better accuracy for our neural network, we evaluated a variety of neural networks by adjusting the number of layers and neurons to determine the optimal architecture for our neural network. Upon analyzing the mean squared error loss function, six layers neural network along with thirty neurons are selected as the final form. Thereafter, the trained neural network is validated by comparing the neural network outputs to those of numerical simulations. These results validate the PINN's reliability in predicting flow fields, which is determined to be

roughly 1% percent for velocity in the x-direction, 2% for velocity in the y-direction, 0.6% for pressure, and 1.5% for density. Beyond that, it also ensures that the predicted fields fulfill the motion equations. Consequently, we believe that our current work will spark interest in physics-informed deep learning, which may be applied successfully in a wide range of mechanics applications.

Acknowledgement

The Doctoral Training Unit Data-driven computational modeling and applications (DRIVEN) is funded by the Luxembourg National Research Fund under the PRIDE program (PRIDE17/ 12252781). https://driven.uni.lu

Chapter 4

Using the trained recurrent neural network to predict particles position for time more than training time[1]

[1] Fateme Darlik and Bernhard Peters. “Using the trained recurrent neural network to predict particles position for time more than training time”. In: (Sept. 2023). Unpublished Manuscript.

Abstract

In this paper, we explore the use of neural networks to predict the motion of particles beyond the training time. To address the challenge of predicting particle behavior beyond the training time, we propose a recurrent physics-informed neural network (PINN) approach that incorporates the underlying physical principles governing the motion of particles in mixing drums, considering both particle-particle interactions and particle-wall interactions.

The proposed methodology utilizes two neural networks, with the first predicting particle's velocity based on time and position inputs, while the second predicts the particle's position for the next time step based on time, position, and velocity inputs. We present the results of two case studies. The first case study involves simulating 444 particles in a moving grate chamber using XDEM software. The neural networks are trained on data collected from 30 to 60 seconds of the simulation, with periodic boundary conditions applied to the chamber. The results show that the neural networks can accurately predict particle positions beyond the training time.

In the second case study, two piles of particles are considered in a rotating drum, and the neural networks are utilized to predict particle positions for times beyond the training time. The results demonstrate that the proposed methodology can effectively predict particle positions for both case studies.

Overall, the paper shows that the proposed PINN approach can be a useful tool for predicting particle motion in mixing drums beyond the training time, and has potential applications in the fields of particle technology and process engineering.

Keywords: Recurrent neural network, Physics-informed neural network, Mixing drum, Moving grate chamber, Particle motion prediction

4.1 Introduction

Numerical simulation techniques (such as discrete element methods) have been used to predict the behavior of particles. The discrete element method (DEM) has been increasingly

used to model particle flow [83, 84, 85]. The main advantage of DEM modeling is the information at the particle scale, such as trajectories of and transient forces acting on individual particles, can be obtained to facilitate the understanding of particle dynamics.

However, despite its advantages, current DEM simulations are often limited due to their computational cost, leading to simplifications in the simulated conditions [86]. These simplifications prevent the accurate representation of real-world industrial processes that operate on larger scales and under more complex conditions [87]. Consequently, there is a need for alternative approaches that can address these limitations and provide more accurate predictions.

In recent years, neural networks are used as a promising alternative for predicting physics phenomena in various fields of science and engineering [8]. Neural networks become increasingly popular due to their ability to handle complex data, non-linearity, and make accurate predictions. However, conventional machine learning methods lack the physical constraints that are necessary for predicting particle motion accurately.

To bridge this gap, the proposed approach combines the strengths of two neural network architectures: the physics-informed neural network (PINN) [25] and the recurrent neural network (RNN) [88]. The PINN integrates physical laws and constraints into the training process, making it suitable for capturing the physics underlying particle dynamics. On the other hand, the RNN excels in modeling time-dependent sequences, enabling the prediction of particle positions over extended periods.

By combining the PINN and RNN, the proposed approach leverages the benefits of both models. The PINN ensures the predictions adhere to physical laws and constraints, while the RNN captures the temporal dependencies in particle positions. This integrated approach allows for accurate prediction of particle motion not only during the training phase but also for periods beyond the training duration, providing a more comprehensive understanding of particle dynamics in complex systems.

We considered the implementation of our methods through the use of two case studies. The case study process will allow us to evaluate the efficacy of our techniques and determine

whether the proposed technique is generalizable across several scenarios rather than being restricted to a particular case. To this end, we have identified two potential case studies: the Biomass chamber and the mixing drum. The primary focus of both case studies was to evaluate the efficacy of the suggested approach in predicting particle positions beyond the training period. The importance of choosing each case will be discussed individually in the following paragraphs.

The reason for choosing the biomass case is the significant role played by biomass energy, which is gradually gaining popularity as a viable alternative to traditional fossil fuels for energy production [89]. Its renewable and sustainable nature makes it a highly attractive option for meeting the energy demands of today's world. The biomass particle motion in a biomass chamber can be influenced by various factors, such as the speed and direction of the moving grate, the size and shape of the particles, and the airflow within the system [90]. Predicting the behavior of these particles can have significant implications in areas such as materials handling, powder processing, and industrial manufacturing [91].

The second case study is mixing drums. In many applications, the mixing process involves the movement of particles within the drum, which can be complex and dynamic. Mixing drums are widely used in industrial processes for mixing and blending powders, liquids, and gases [92]. The efficient and effective mixing of materials is critical for ensuring the mixing performance, and product quality [93].

This study focuses on predicting the position of particles for time duration beyond the training time. While neural networks have been used extensively to predict particle positions during the training period, there are fewer studies on extrapolating predictions beyond this period. This is partly because particle behavior can be highly unpredictable and subject to a wide range of physical factors, making it difficult for the network to generalize beyond the training data. The basic idea is to use a training set of particle positions over time to train the network, and then use the network to predict future positions beyond the time range of the training set.

Generally speaking, researchers are actively working to develop more advanced neural

network architectures that can better handle extrapolation tasks. For example, recurrent neural networks (RNNs) are particularly effective in capturing time-dependent behavior and making predictions beyond the training period. RNNs are well-suited to predicting time-series data, as they can remember past inputs and use that information to predict future outputs. Overall, the use of neural networks for predicting the position of particles for time duration beyond the training time has the potential to reconstruct the field of particle physics.

4.2 DEM simulation for data generation

In this study, two case studies are used to develop and validate our proposed methodology. Both case studies are composed of two packs of particles that are solved using discrete element methods (DEM). Due to the unavailability of experimental data, the simulation data is used to train and evaluate the effectiveness of the neural network. The simulation is done using the eXtended discrete element method (XDEM) [72] which is an in-house code that implemented DEM. Not only does XDEM provide information regarding the position and orientations of the particle, but also it provides some information regarding internal temperature, porosity distribution, and mechanical impacts [94].

4.2.1 Biomass chamber

The selection of this case study stems from our previous research [13, 14], which provide readers with extensive insights into the fundamental physics underlying this specific problem. In this current study, we employ XDEM to simulate the motion of particles on a moving grate. The particles enter the domain from the right side and exit from the left side as they traverse through the system. To ensure a constant number of particles during the simulation, we apply periodic boundary conditions at both the input and output boundaries of the domain. This means that particles exiting the domain through the output gate are re-entered into the chamber, allowing for a continuous flow and maintaining a steady particle population within

the system. The total number of particles in the system is 444, all with a uniform radius of $1.5cm$ and a density of $1197kg/m^3$. At the initial state, the particles in the system exhibit a stationary behavior. Subsequently, as a result of the motion of the grate and the influence of gravity, the particles undergo dynamic motion and initiate movement.

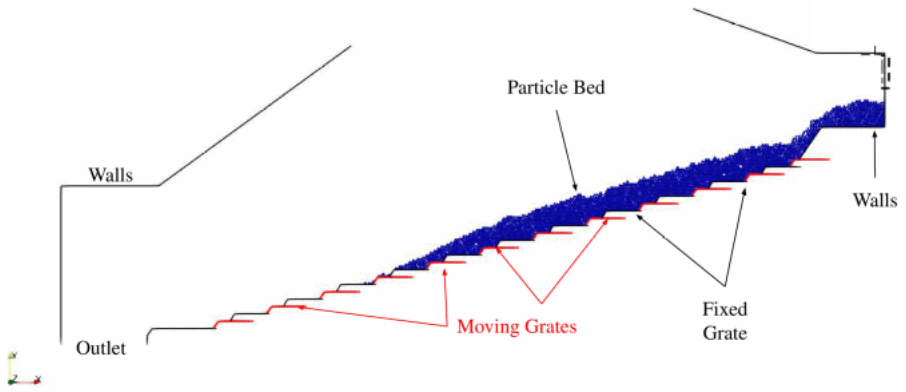


Figure 4.1: XDEM Biomass furnace Case setup [13]

The modeling of particles involves representing them as compressible spheres that interact with an interaction model inspired by the work of Cundall and Strack [95]. In this model, the normal force F_n acting on the particles is composed of a linear spring force and a dashpot force. The spring force creates an elastic repulsive force, while the dashpot force contributes damping, as described by the following formula:

$$\overrightarrow{F}_n = -\left(k_n \delta_n + c_n \delta_n\right) \tag{4.1}$$

The dynamics module of XDEM software predictions particle position, velocity, and acceleration. It uses a soft sphere model, where particles are assumed to be deformable and can overlap each other. The interaction particle contact forces are computed through the magnitude of overlap between the particles, based on the force-displacement law chosen. The particle hardness is described with Young's modulus, whereas the particle energy dissipation is modeled with a dampener and/or dash-pot. The transnational and

rotational movements of individual particles are tracked using the classical mechanics' equations as described in Eq 5.1 and Eq 5.2 respectively.

$$m_i \frac{d\vec{v_i}}{dt} = m_i \frac{d^2\vec{X}_i}{dt^2} = \vec{F_i^c} + \vec{F_i^g} + \vec{F_i^{ext}} \tag{4.2}$$

$$I_i \frac{d\vec{\omega_i}}{dt} = \sum_{j=1}^{n} \vec{M}_{i,j} \tag{4.3}$$

where $\vec{F_i^{ext}}$ is the sum of all the external forces acting on the particle, such as buoyancy forces $\vec{F_B}$ and drag forces $\vec{F_D}$.

4.2.2 Large drum - Mixing process

In the current case study, the mixing process of two piles of particles in the mixing Drum is studied 4.7. The two piles of particles are assumed to have the same properties, volume, and density. The material softwood is chosen as the material for the particles. The drum had an angular velocity of $0.1\ rad/s$ in the z-direction. At the starting point, all the particles are stationary. Afterward, the drum started to rotate, and the particle mixing process began. After passing the transient state, a complete mixing process of the piles is taken place.

The dynamics impact model employed in this study is based on the Hertz-Mindlin theory [96]. The Hertz-Mindlin model combines the principles of Hertz theory, which describes the normal elastic force between contacting bodies, with Mindlin's research on energy dissipation during normal contact. The normal force in the Hertz-Mindlin model is defined as follows:

$$\overrightarrow{F_n} = -(\frac{4}{3} E_{ij} \sqrt{R_{ij}} \delta^{\frac{3}{2}} + c_n \delta^{\frac{1}{4}} \dot{\delta}) \tag{4.4}$$

The normal dissipation coefficient is formulated following the proposal by Tsuji et al. [97].

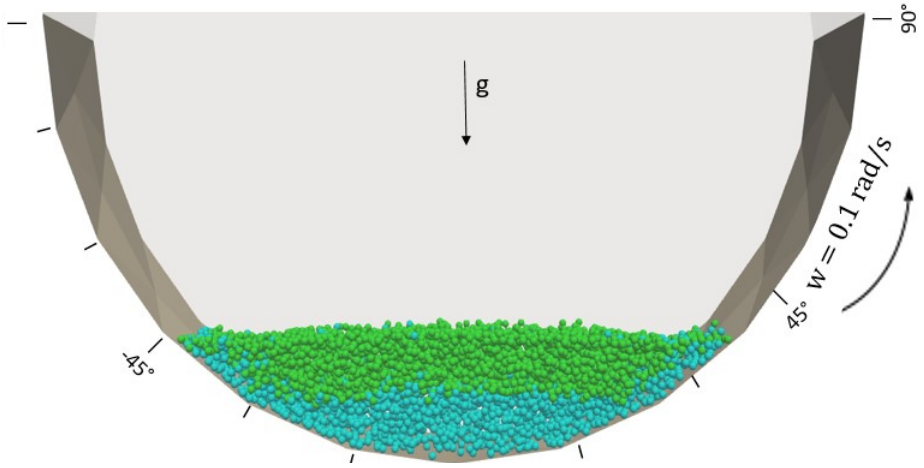


Figure 4.2: Two piles of particles in mixing drum with the angular velocity of $0.1\ rad/s$

$$c_n = \ln e_n \sqrt{\frac{5 m_{ij} k_n}{\pi^2 + \ln e_n^2}} \tag{4.5}$$

$$k_n = \frac{4}{3} E_{ij} \sqrt{R_{ij}} \tag{4.6}$$

The tangential forces encompass both static and dynamic friction components, which can be expressed as follows.

$$\overrightarrow{F}_t = min(k_t \delta_t + c_t \dot{\delta}_t, \mu \overrightarrow{F}_n) \tag{4.7}$$

The expression for the tangential stiffness is as follows:

$$k_t = 8 G_{ij} \sqrt{R_{ij} \delta_t} \tag{4.8}$$

The effective shear modulus is defined as:

$$\frac{1}{G_{ij}} = \frac{2 - \nu_i}{G_i} + \frac{2 - \nu_j}{G_j} \tag{4.9}$$

The tangential dissipation coefficient c_t is correlated with the tangential coefficient of restitution by the following formula:

$$c_t = \ln e_t \sqrt{\frac{5(4m_{ij}k_t)}{6(\pi^2 + \ln e_t^2)}} \tag{4.10}$$

The shape of each particle is considered a sphere and the radius is set to $2cm$. The properties of the particles can be found in the table 4.1.

Table 4.1: Particle Parameters for Each Pile

Parameter	Value
Number of particles	7626
Mass density [kg/m^3]	400
Young modulus [Pa]	$20e8$
Poisson's ratio	0.28
Restitution coefficient	0.1
Viscosity	$2.42e6$
Radius [cm]	2

The rotation of the drum causes the mixtures of the particles. In figure 4.3, the snapshots of the particles in different time steps are illustrated.

The mixing process of particles in a drum begins with the particles initially settling at the bottom. As the simulation starts and the drum undergoes angular motion, the particles begin to move in the direction of the drum's motion. Over time, the particles that rise up will fall onto the other particles, repeating this process will lead to complete mixing. It is worth noting that after a certain period, the particles will tend to settle in specific regions, leading to a semi-steady mixing process.

As depicted in the figures, at 10 and 20 seconds, the two distinct piles of blue and green particles are still distinguishable, indicating incomplete mixing. However, at 40 seconds and beyond, such as 80 seconds, the particles exhibit a well-mixed state where the different piles are thoroughly blended together.

This observation highlights the temporal aspect of the mixing process, where sufficient time is required for the particles to interact and redistribute themselves, ultimately achieving a homogeneous mixture. The figures visually demonstrate the progression of mixing over time, showing the evolution from initial segregation to eventual mixing.

The figures and presented below illustrate the mean velocity of particles in each pile over time. Upon examination, it becomes evident that at the initial time, the mean velocity of the particles has fluctuations. However, as time progresses, these fluctuations diminish, and the particles converge toward a relatively constant mean velocity. Notably, this convergence occurs for both piles and interestingly, it aligns with the time frame of 40 seconds when full mixing is observed in the .

The depicted trend suggests that as the particles undergo mixing, their velocities become more consistent and approach a stable value. The reduction in velocity fluctuations signifies a relatively homogeneous mixture.

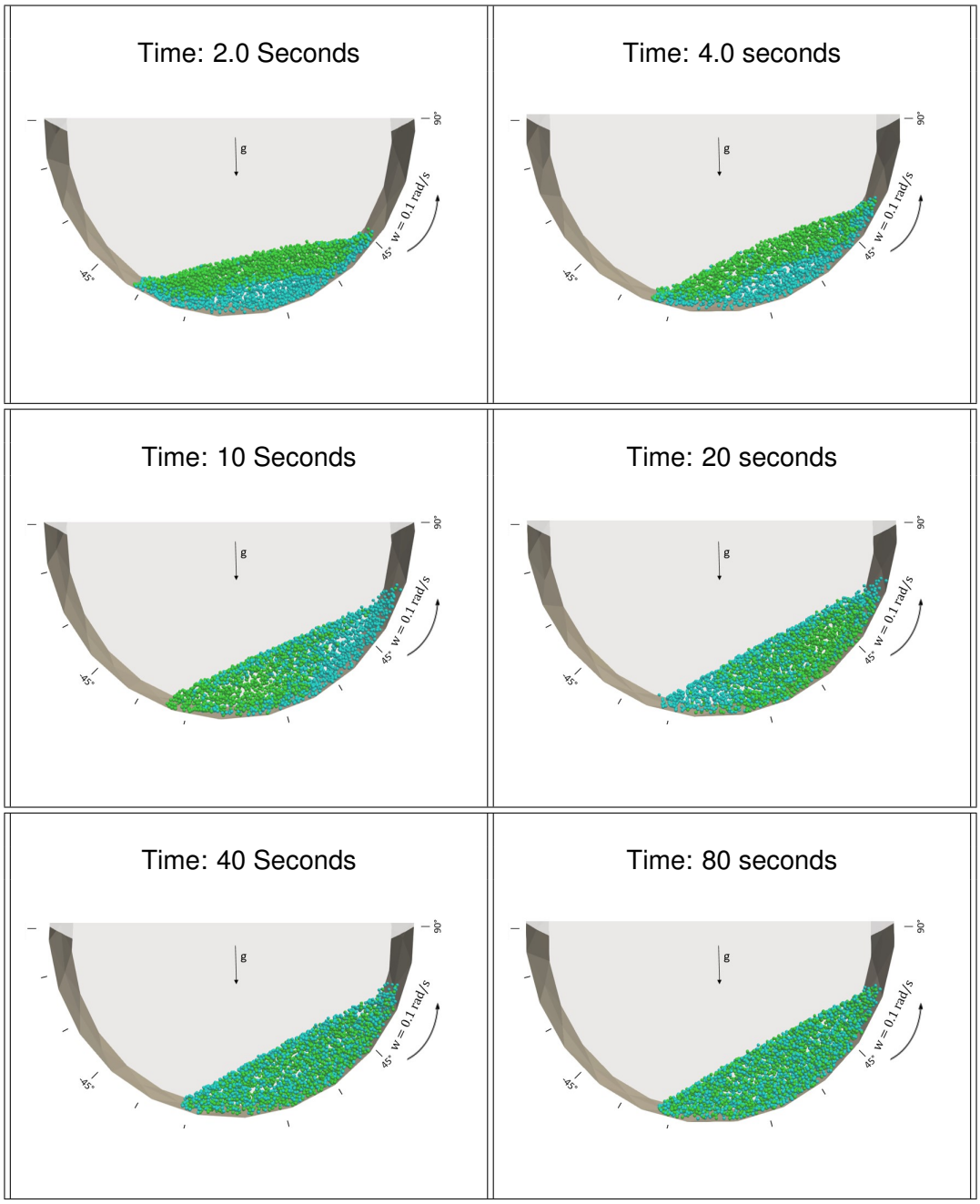


Figure 4.3: Particles in mixing drum

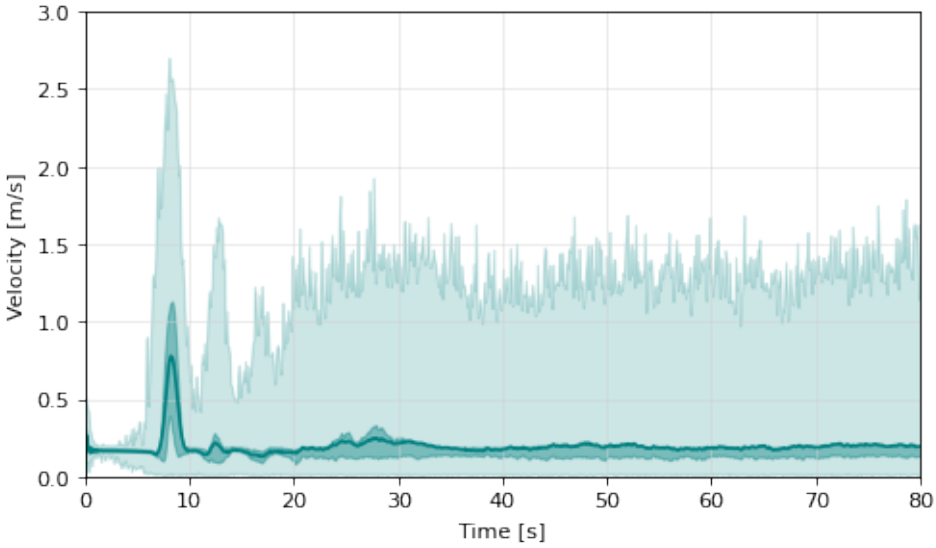


Figure 4.4: The average of velocity magnitude of the first pile of particles in time

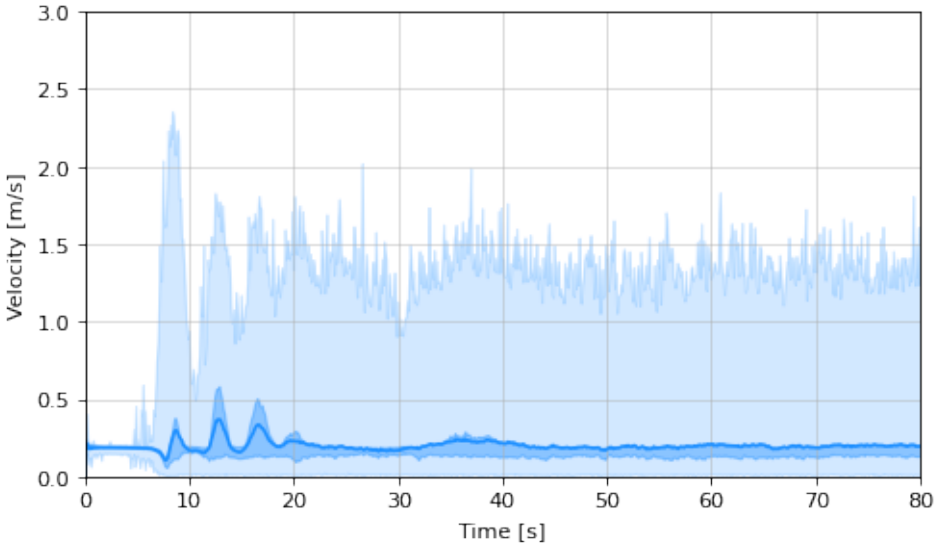


Figure 4.5: The average velocity magnitude of the second pile of particles in time

4.3 Neural Network

4.3.1 Recurrent neural network

A neural network can be used as a tool to help solve engineering problems. With the help of Neural networks, we can avoid time-consuming processes in engineering simulation, and accelerate the simulation, saving much time [98, 99]. Also, the progression of using neural networks in engineering problems is progressing day by day, but it has been shown that it can have intriguing results. Additionally, nowadays discovery divided the neural network into different kinds of categories, which is one of the common ones, which is dividing the neural network into two categories supervised and unsupervised learning. Which should be chosen related to the problem. In this investigation, because of the nature of the problem, we went through supervised learning. Also, neural network layers can be organized differently and will be brought out different kinds of neural networks such as fully connected neural networks, convolutions neural networks, Recurrent neural networks, and so on.

Neural networks are used as a tool to help solve engineering problems. The neural networks, help the time-consuming processes in engineering simulation.[98, 99]. Presently, there are multiple categories of neural networks available, offering diverse capabilities and applications. Some of these include fully connected neural networks, convolutional neural networks (CNNs), recurrent neural networks (RNNs), and several other variants. Each type of neural network possesses unique characteristics and functionalities, making them suitable for various tasks and domains. The continuous development and exploration of different neural network architectures have expanded the possibilities and effectiveness of artificial intelligence in solving complex problems and improving various fields of study, such as image recognition, natural language processing, time series analysis, and more.

RNNs are particularly adept at handling sequential or time series data, making them well-suited for tackling temporal problems. Their practicality in handling such data has led to widespread utilization in diverse applications, including language translation, natural language processing (NLP), speech recognition, and image captioning. This popularity

stems from the RNN's ability to capture and learn from the sequential data, enabling them to generate meaningful and accurate predictions or outputs in these domains.

Generally speaking, mechanical engineering problems are mainly time-dependent's problems, with specific time intervals. For this particular problem, the simulation outcomes at time step n are determined based on the results obtained from previous time steps (specifically time step $n-1$). Consequently, it is essential to establish an initial condition as a prerequisite for solving such problems.

Recurrent neural networks transform a vector of hidden states, d, from time t_{-1} to time t, in the following way:

$$h_t = f(h_{t-1}, x_t) \tag{4.11}$$

In this context, let t denote the time, h_t and h_{t-1} represent the sequential states, x_t represents the input variables, and f denotes the activation function applied to the hidden state. By using the non-linear activation function $tanh$, the equation (4.11 has a following format:

$$h_t = tanh((W_{hh}h_{t-1} + B_{hh}) + (W_{xh}x_t + B_{xh})) \tag{4.12}$$

The states within the cells of recurrent neural networks undergo iterative updates by applying the transformation $f(dh_t, x)$. In its basic form, $f(dh_t, x)$ can be implemented as a single-layer perceptron or a fully-connected dense layer. However, there are more advanced design options available for constructing the cell architecture. Among the various recurrent neural network architectures proposed by researchers, the Long Short-Term Memory (LSTM) cell has gained significant recognition and is widely adopted.

The concept behind LSTM presents a distinctive approach compared to conventional RNNs, introducing the notions of "long-term memory" and "short-term memory". In this methodology, the network's connection weights and biases dynamically evolve during each training episode, allowing the network to adapt and learn from the data. At the same time, the activation patterns within the network continually update with each time step, capturing the temporal dynamics of the input sequence. The fundamental goal of the LSTM

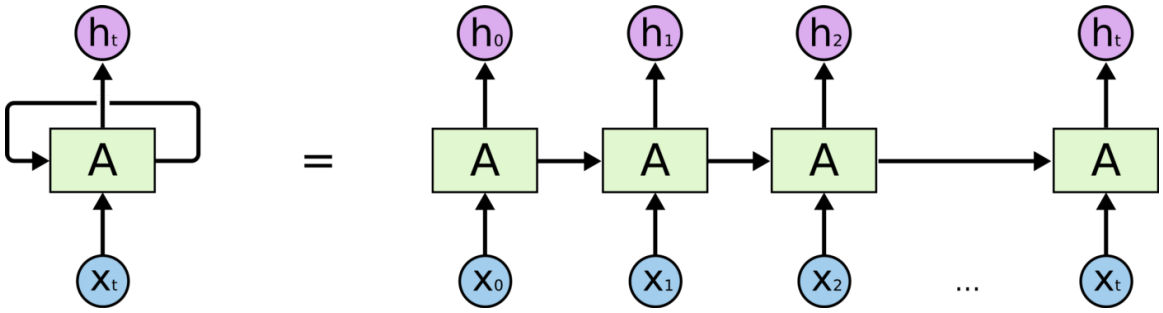


Figure 4.6: Long short-term memory

architecture is to equip RNNs with a robust short-term memory capability that can span an extensive range of timesteps.

A typical LSTM unit comprises a cell, an input gate, an output gate, and a forget gate. The cell retains values across arbitrary time intervals, while the three gates control the flow of information into and out of the cell [100].

LSTM networks excel in tasks involving the classification, processing, and prediction of time series data. This is especially relevant in scenarios where significant events within a time series can be separated by unknown time intervals. LSTMs were specifically designed to address the issue of vanishing gradients that can hinder the training of traditional RNNs. Within the equations of LSTM 4.13 to 4.17, one can observe the expressions that establish the relationship between the input and output layers within each LSTM cell. These equations define the transformations and computations performed to process the input data and produce the corresponding output.

$$f_t = \sigma_g(W_f x_t + U_f c_{t-1} + b_f) \tag{4.13}$$

$$i_t = \sigma_g(W_i x_t + U_i c_{t-1} + b_i) \tag{4.14}$$

$$o_t = \sigma_g(W_o x_t + U_o c_{t-1} + b_o) \tag{4.15}$$

$$c_t = f_t \odot c_{t-1} + i_t \odot \sigma_c(W_c x_t + b_c) \tag{4.16}$$

$$h_t = o_t \odot \sigma_h(c_t) \tag{4.17}$$

The subscript t signifies the time step. The input vector at time step t is denoted as x_t. The forgotten gate's activation function is given by f_t, the input's activation function is represented by i_t, and the output's activation function is denoted as o_t. The hidden state is represented by h_t, while the cell state is denoted as c_t. The parameters to be learned during training, including the weight and bias, are collectively represented by W.

In terms of activation functions, we have the sigmoid function denoted as σ_g, the hyperbolic tangent function denoted as σ_h, and another hyperbolic tangent function denoted as σ_c. These activation functions play a crucial role in processing and transforming the input data within the LSTM unit.

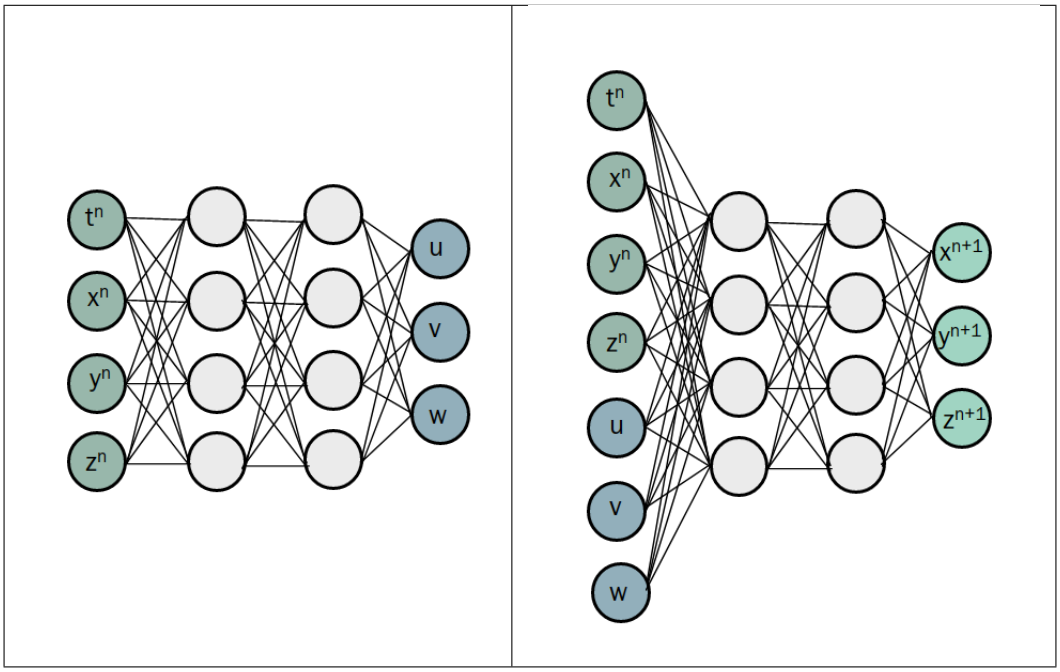


Figure 4.7: Particles in mixing drum

4.3.2 Activation function

The role of the activation function in the neural network is to compute the network's output [101], with the function mapping the resulting values within a specific range, such as between 0 and 1, or -1 and 1, depending on the specific activation function employed.

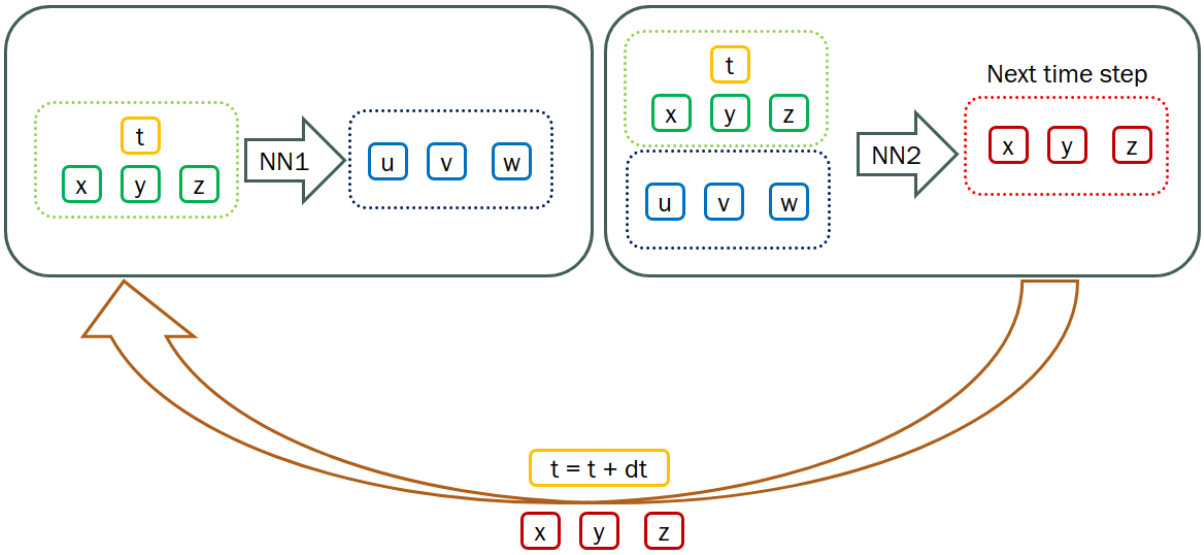


Figure 4.8: Long short-term memory architecture

The activation functions are divided into two main types linear activation functions, and non-linear activation functions. In the hidden layers of a neural network, it is common to use a differentiable nonlinear activation function. This choice enables the model to acquire the ability to learn complex models that would be challenging for a network trained solely with a linear activation function. The sigmoid loss function is one of the l

In this paper, the sigmoid function is used as an activation function. The sigmoid function, also known as the logistic function, is commonly employed in neural networks due to its ability to introduce non-linearity into the model. It maps the input values to a range between 0 and 1. The equation of the sigmoid activation function is given by:

$$\sigma(x) = \frac{1}{1 + e^{-x}} \tag{4.18}$$

where x represents the input to the sigmoid function. Additionally, the hyperbolic tangent (tanh) function is utilized as an activation function. in the current study. The tanh function is commonly employed in neural networks for its ability to introduce non-linearity and capture a broader range of input values. It maps the input values to a range between -1 and 1,

making it suitable for tasks that involve capturing both positive and negative patterns. The equation for the tanh activation function is as follows:

$$\tanh(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}} \tag{4.19}$$

where x represents the input to the tanh function.

4.3.3 Batch size and number of epochs

Efficient training of a model requires the proper configuration of various parameters, one of which is the batch size. The batch size refers to the number of samples processed before updating the model. Similarly, the number of epochs represents the complete passes made through the training dataset. It is crucial to set an appropriate batch size that meets specific criteria: it should be greater than or equal to one and less than or equal to the total number of samples in the training dataset [102].

4.3.4 Loss function

The loss function plays a vital role in evaluating the accuracy of the neural network's prediction output and guiding the updates of its weights and biases in each epoch. There are various types of loss functions designed to suit different applications of neural networks, broadly categorized into Binary Classification Loss Function and Regression Loss Function.

When assessing our problem, which does not involve classification tasks, we focus on the regression loss functions. One commonly used loss function in regression tasks is the Mean Squared Error (MSE). The MSE is a straightforward and widely taught loss function in introductory Machine Learning courses. It calculates the squared difference between the model's predictions and the expected outputs, averaging this value across the entire dataset [103].

However, given the context of our application, we decided to incorporate a physics-informed loss function. This choice enables us to go beyond solely assessing the disparity between

the neural network's predictions and the ground truth values using mean squared error(MSE) but also ensures that the predicted values conform to the fundamental laws of physics governing the problem.

In our specific scenario, the neural network is employed to predict particle velocities, based on the inputs of position and time. As a result, the physics-informed loss function serves the dual purpose of not only quantifying the discrepancy between the predicted and actual values but also validating that the predicted values satisfy the inherent relationship between velocity and position. This approach enables us to align the neural network's outputs not only with the observed data but also with the underlying physical principles that govern the system.

The physic-informed mean squared error is defined by the following equation:

$$MSE := \frac{1}{N}\sum_{i=1}^{N}(u_i - \hat{u}_i)^2 + (e(u_i))^2 \tag{4.20}$$

$$e_1 = u - \frac{dx}{dt}, e_2 = v - \frac{dy}{dt}, e_3 = w - \frac{dz}{dt} \tag{4.21}$$

Where N is the number of samples we are testing against, and u_i and $\hat{u}_i$ are the actual and predicted value repetitively. The e_i are the equations, x, y, z are the positions, and u, v, w represents the velocities.

The neural network architecture, including the specific type of network, choice of the loss function, activation function, and batch size, has been established. With these components defined, the neural network is now prepared to be applied to the available datasets for further analysis and processing. The neural network architecture, including the specific type of network, choice of the loss function, activation function, and batch size, has been established. With these components defined, the neural network is now prepared to be applied to the available datasets for further analysis and processing.

4.4 Results and discussions

4.4.1 Case study 1: Moving grate chamber

This section presents the results of the first case study. As mentioned in the previous section, in this study a neural network consisting of two neural networks is proposed. The first neural network trained with the time and position of the particles as its inputs and predicted the velocity of the particles as its outputs. The second neural network, on the other hand, takes as inputs the time, position, and velocity of the particles and predicts the next time step's positions of the particles. Both of these neural networks are trained on the datasets of the particles' motion in the moving grate chamber.

In this study, a total of 444 particles, having a radius of $1.5cm$ and density of $1197kg/m^3$, are simulated within a moving grate chamber. In this simulation, a periodic boundary condition is defined on both sides of the geometry (input and output) to ensure that the whole number of particles within the domain remains constant. In other words, any particle that exits through the output gate is re-entered through the input gate. The simulation was executed for a duration of 60 seconds, using a time step of 0.001. To avoid any initial fluctuations or instability of the simulation, and also to achieve a semi-steady state, the first 30 seconds of the simulation are discarded, and the dataset is collected from the remaining 30 to 60 seconds of the simulation. The dataset is generated by running the simulation using XDEM.

The available dataset is used for training the first neural network, which consists of seven hidden layers, each containing 100 neurons. The ReLU activation function is applied to the hidden layers, while the output layer uses the Sigmoid activation function. The Mean Squared Error is selected as the loss function, and the ADAM is used as the optimizer. In order to mitigate the risk of divergence in the neural network resulting from variations in input and output variable ranges, a normalization process is employed, where all variables are scaled to a standardized range of 0 to 1. Figure 4.9 shows the mean squared error loss function for the first neural network, indicating an error of 7×10^{-5}.

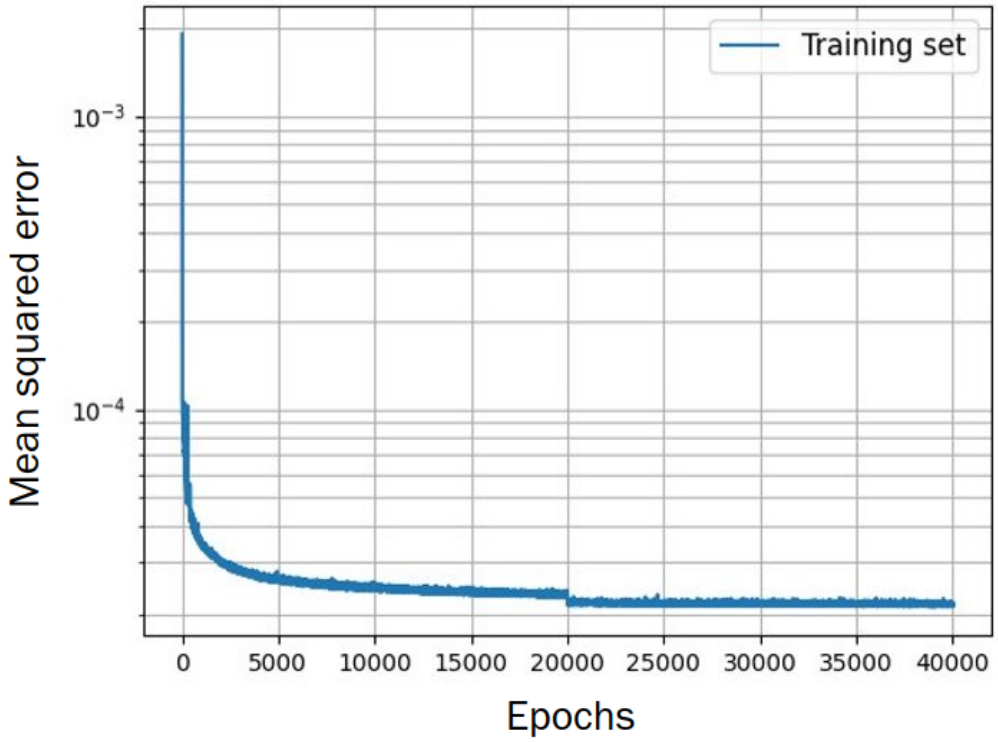


Figure 4.9: First neural network's loss function error on the first case study (Moving grate)

The second neural network was also trained on the datasets, using the same neural network features including the same number of hidden layers, neurons, activation functions, loss function, normalization range, and optimization method as the first neural network. Figure 4.10 displays the mean squared error loss function value, which is 9×10^{-6}. By comparing figure 4.9 and figure 4.10, it is evident that the second neural network has a lower loss value than the first one. It indicates that the second neural network is more proficient in predicting the outputs of the neural network. This can be attributed to the fact that the outputs of the second neural network (which is the next time step's position of the particles) have the same type as the inputs (position and velocity). This characteristic enables the neural network to establish a stronger relationship between the outputs and inputs compared to the scenario where the inputs consist solely of position and the outputs consist solely of velocity.

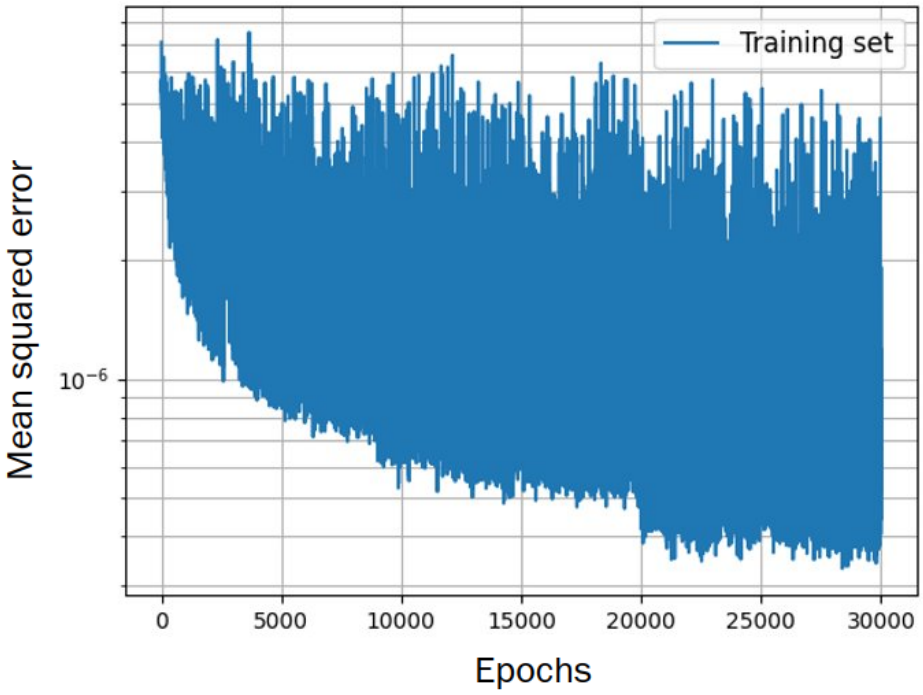


Figure 4.10: Second neural network's loss function error on the first case study (Moving grate)

After training both neural networks, they will be used to make predictions for particle positions beyond the training time.

The prediction for the period beyond the training time is conducted for a duration of 200 seconds (the process is demonstrated in figure 4.8). The prediction started from the final training period's time step, which, in this case study is at 60 seconds. The particle's position at that time was inputted into the first trained neural network, which generated a prediction of the particle's velocity. Subsequently, the predicted velocity addition to the particle's positions is inputted into the second neural network to generate the next time step's particle's position. This process will be performed continuously throughout the entire duration of the simulation period being considered.

The outcomes for four different time points, i.e., 60, 100, 150, and 200, are displayed in figure 4.11. Figure 4.11 demonstrates that the predictions of the particle's position beyond

the training time have a reasonable pattern and are physically plausible. It is shown that the particles remained on the grate without exhibiting any unexpected behaviors such as floating or moving upward. Therefore, this simulation can be regarded as an approximate representation of the actual simulations.

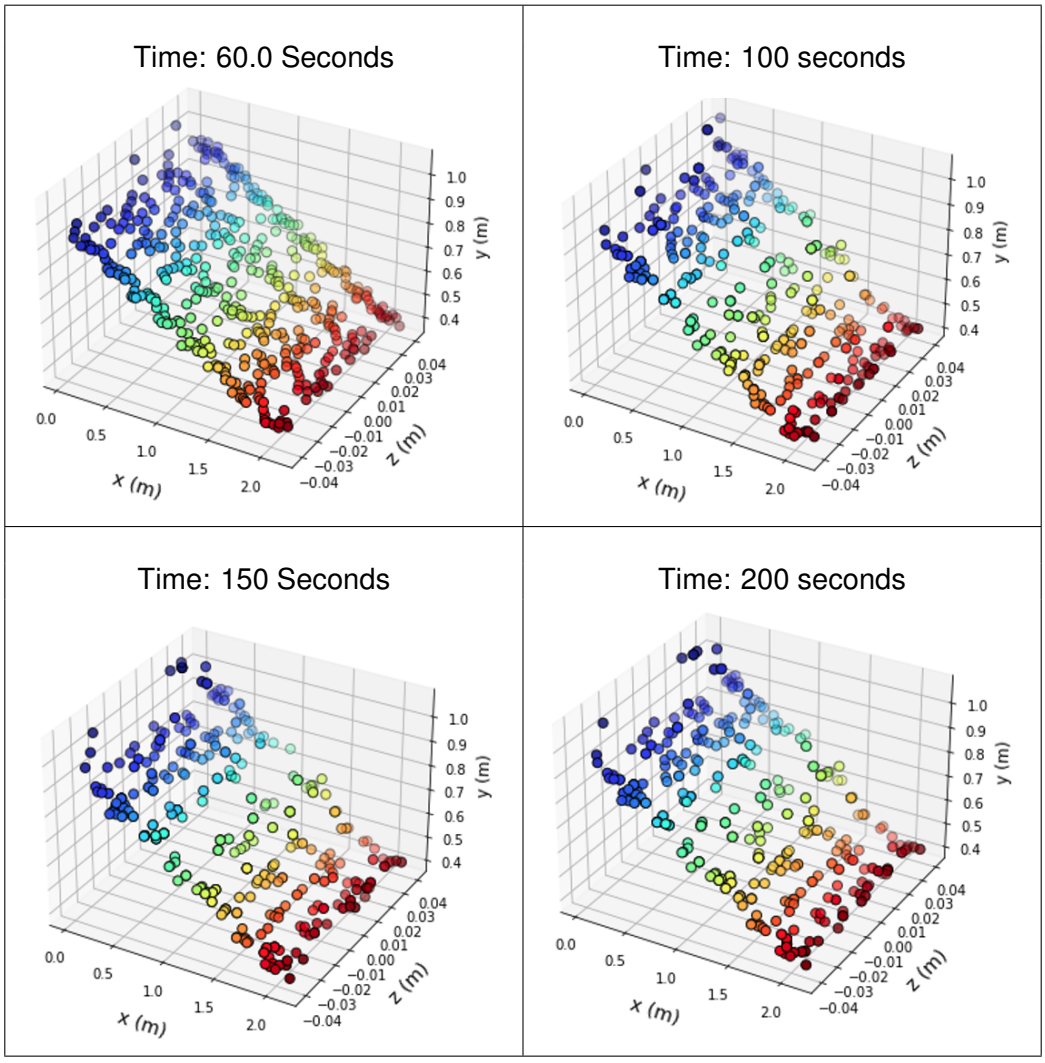


Figure 4.11: Particles in mixing drum

The 2-dimensional diagrams of the particle's position at 200 seconds in both the XY direction and ZX-direction are illustrated in Figure 4.12. Based on the illustrations, it can be inferred that the particles are evenly distributed in both the XY direction and ZX direction.

Even after predicting for a time that is four times longer than the training time (i.e., the training time is 30 seconds, the prediction time is 200 seconds, which is 140 seconds beyond the training time), the distribution of particles is still present, and there is no evidence of particle aggregation.

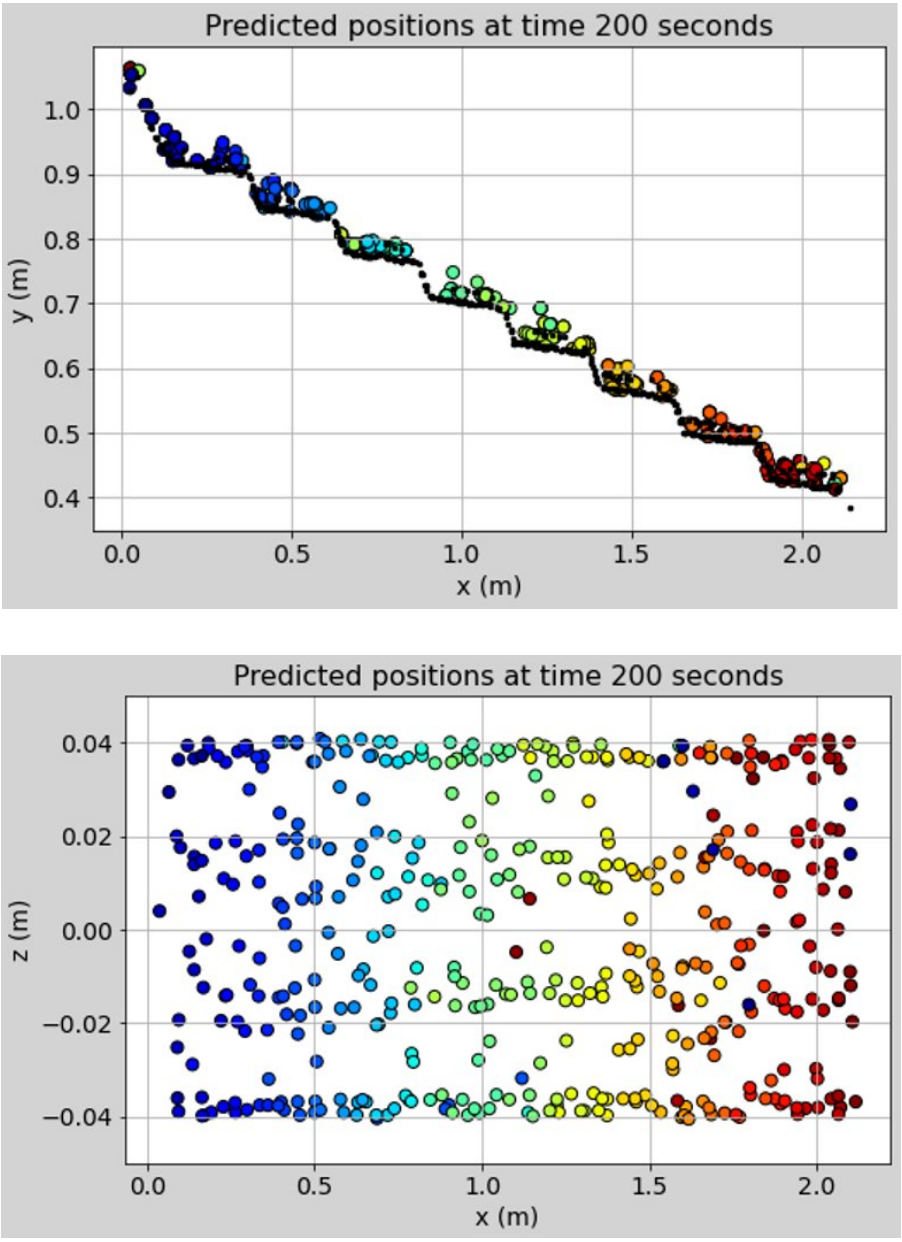


Figure 4.12: Particles in mixing drum

4.4.2 Case study 2: Mixing Drum

In this section, the efficacy of the proposed combination of neural networks is evaluated through its application to a new case study. The new case study is the mixing process of particles in a rotating drum. This simulation includes two piles of particles in the drum, where each consisting of 7600 particles with a $2cm$ radius and a density of $400kg/m^3$. The angular velocity of the drum is set to $0.1rad/s$. In this model, the interaction forces between particles and between particles and the drum are considered, and the Hertz-Mindlin model is used to model the interaction force. The simulation period lasts for a total of 140 seconds. To obtain a semi-steady state during the simulation, the initial time period is discarded.

In order to achieve precise particle position predictions, the training period for the neural network model is extended beyond the time required for the drum to complete a single revolution. This strategy enables the model to capture the varying particle movement patterns throughout different stages of the mixing process, consequently enhancing the accuracy of the predictions. Therefore, the training period commences at 80 seconds and extends until 140 seconds. By adopting this approach, the neural network becomes more adept at extrapolating particle positions beyond the training time, thereby enhancing the overall model accuracy.

In the second case study, the initial neural network is trained on the dataset. After tuning the neural network's hyperparameters to prevent both over-fitting and under-fitting, the final neural network's architecture is selected. The neural network has ten hidden layers with 100 neurons per each hidden layer. The mean squared error is used as the loss function, and Adam optimization is selected as the optimization method. As with previous sections, all variables are scaled to a range of 0 to 1. It is worth noting that normalization helps in improving the model's performance and allows for a better comparison of the input variables. To assess the performance of the neural network, the average mean squared error (MSE) values for the particle velocities over time are depicted in Figure 4.13. The results indicate that the average MSE values for all three velocities consistently remained below 1.4×10^{-3}. This noteworthy low value demonstrates the neural network's ability to accurately predict

particle velocities.

$$MSE^t := \frac{1}{N}\sum_{i=0}^{N}(y_i^t - \hat{y}_i^t)^2 \tag{4.22}$$

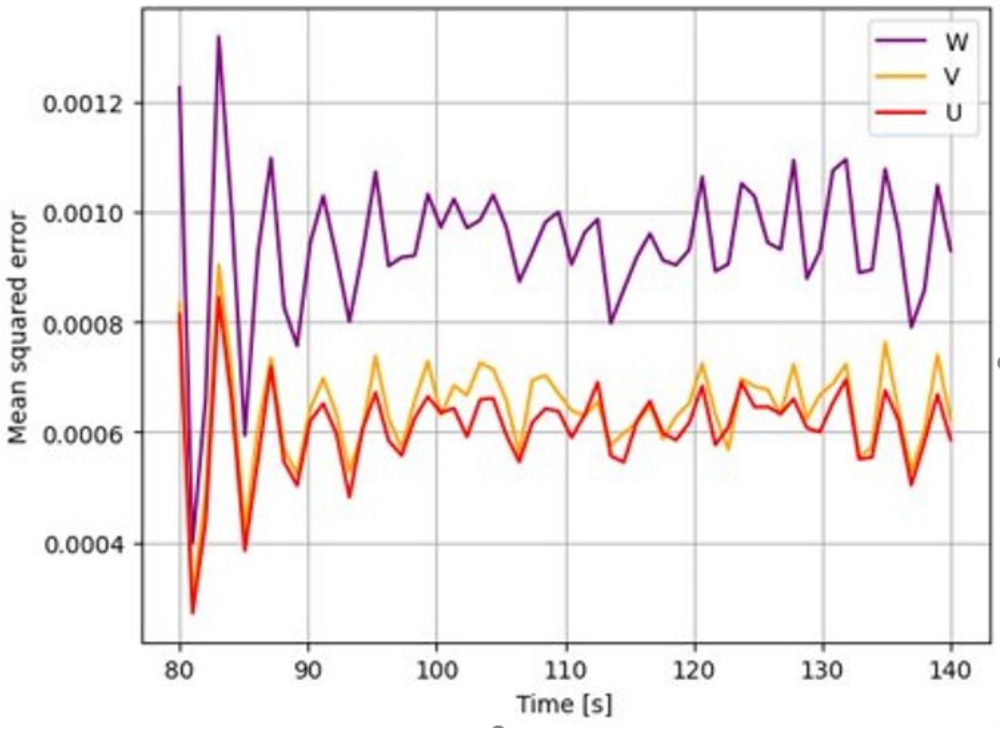


Figure 4.13: First neural network's mean squared error of the second case study (Mixing drum)

Furthermore, to evaluate the accuracy of the predictions made by the first neural network, the average relative percent error (RPE) of the particle's velocity is calculated and plotted in figure 4.14. The results indicate that the relative percent error of all variables is less than 5 percent. This finding indicates a high degree of agreement between the predicted values and actual values, and also are indicative of the effectiveness of the neural network in capturing the underlying patterns in the data and making accurate predictions.

$$RPE^t := \frac{1}{N}\sum_{i=0}^{N}\frac{y_i^t - \hat{y}_i^t}{y_i^t} \tag{4.23}$$

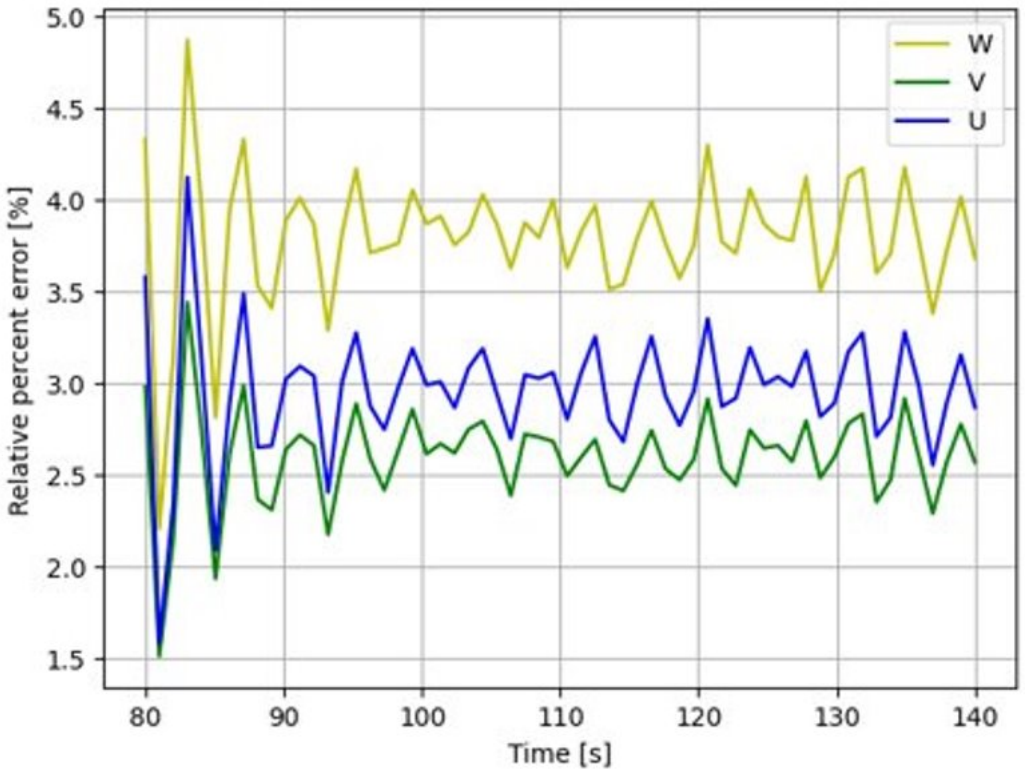


Figure 4.14: First neural network's average relative percent error of the second case study (Mixing drum)

A noticeable disparity in the loss values can be observed between the initial time and the final time of the training, as depicted in figure 4.14. The high value observed during the initial stages of training can be attributed to the network's limited familiarity with the underlying patterns of particle motion. However, as the training progresses, the network gradually acquires a better understanding of these patterns, leading to a reduction in the loss value.

It is worth noting that the use of a recurrent neural network (RNN) in this study allows the network to have the memory of the previous time steps. This can significantly help in training the model to make accurate predictions. This is particularly important in the case of particle motion, where the behavior of the particles in the current time step is dependent on their position and velocity in the previous time steps. Therefore, the RNN architecture helps to capture these dependencies and ultimately leads to better prediction performance.

The observed difference in the loss value of the velocity in the Z direction compared to the velocity in the X and Y directions can be attributed to the complexity of the particle interactions that occur in the Z direction. The particles in the X and Y directions follow a more predictable pattern due to the constant angular velocity of the drum. However, in the Z direction, the particle behavior depends on various factors such as their position and the interactions between them, which may lead to sudden changes in their velocity. Hence, predicting the velocity in the Z direction can be more challenging for the neural network, resulting in a higher loss value. This behavior is evident in both figure 4.14 and figure4.14.

It is crucial to highlight that, upon analyzing Figure 4.14 and Figure 4.14, the loss values for velocity in the X and Y directions demonstrate similar ranges. This observation can be attributed to the fact that the motion of particles in both the X and Y directions is influenced by the angular velocity of the drum. The drum's angular velocity affects both velocities concurrently, leading to a consistent prediction by the neural network and resulting in comparable loss values.

The second neural network undergoes training using the mixing drum dataset. It possesses an identical neural network architecture to the first neural network, including the number of hidden layers, number of neurons per layer, loss function, normalization range, and optimization method.

To assess the performance of the neural network, both the average mean squared error and the average relative percent error of the particle's position over time are presented in figure 4.15 and figure 4.16, respectively.

Figure 4.15 presents the average mean squared error (MSE) values, where the maximum value observed in this plot is 1×10^{-7}. Furthermore, figure 4.16 showcases the average relative percent error (RPE) values, all of which remain below 0.6 percent. These findings validate the high accuracy of this neural network in predicting particle positions throughout the mixing process.

Significantly, it is worth highlighting that the loss values associated with the second neural network are consistently lower in comparison to those of the first neural network.

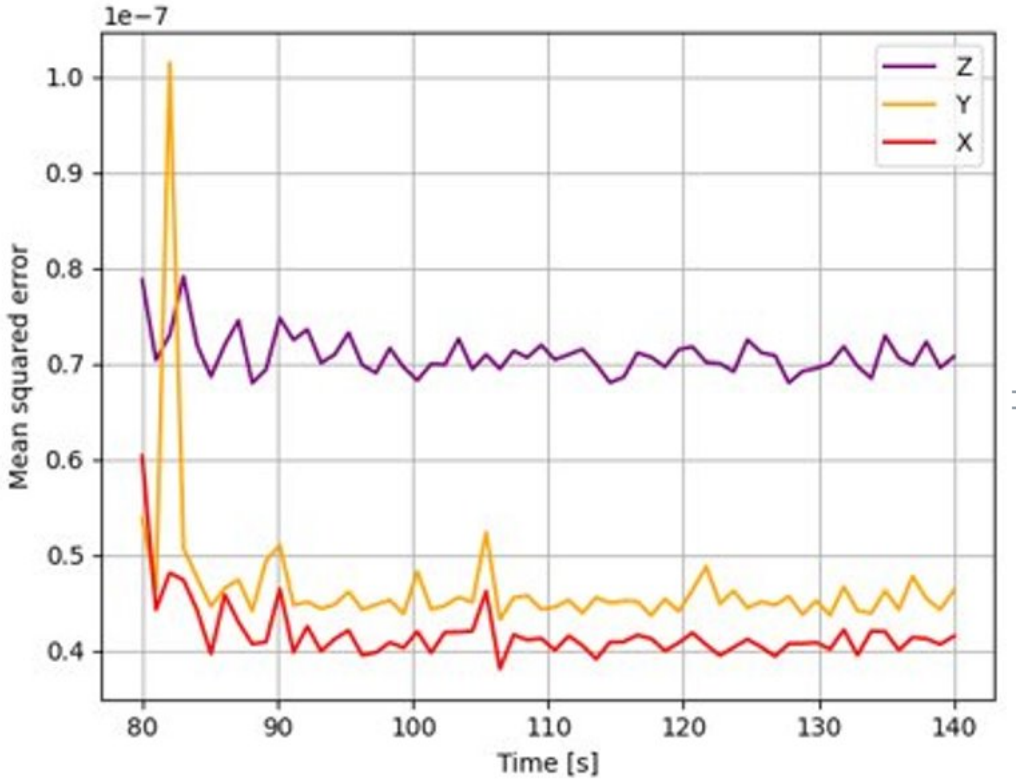


Figure 4.15: Second neural network's mean squared error of the second case study (Mixing drum)

This disparity can be attributed to the compatibility between the outputs of the second neural network (positions) and its inputs (position and velocity). Conversely, the inputs (positions) and outputs (velocities) of the first neural network originate from distinct types, potentially accounting for the marginally higher loss values observed.

The trained neural networks for both cases are saved for future use and are loaded to be used in the predictions for time beyond the training time. The results of the predicted particle positions are shown in figure 4.17. The prediction period extends up to three times the duration of the training period. The predicted particle positions for specific times, including 140, 150, 170, 190, 250, and 300 are displayed in the same figure (figure 4.17 . As can be seen in figure 4.17, the predicted positions of the particles show a physically acceptable distribution. The particles are not randomly aggregated and do not appear to gather on each

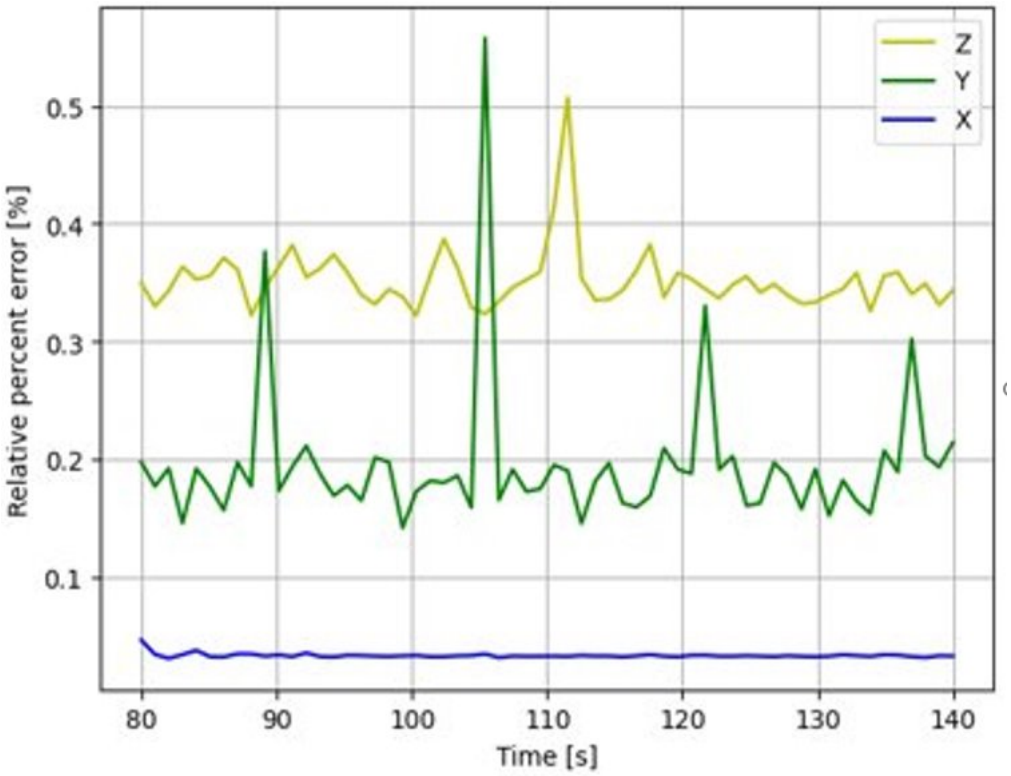


Figure 4.16: Second neural network's average relative percent error of the second case study (Mixing drum)

other, which would indicate unrealistic clumping behavior. Moreover, the particles do not perform upward movement, which would be inconsistent with gravity. Overall, the predicted particle positions appear to be reasonable and in agreement with physical expectations.

As the simulation progresses, the snapshots of particle positions at different time steps exhibit variation. The distribution of particles tends to occur in a dispersed manner, although it remains physically plausible. This limitation can be addressed by increasing the duration of the training period. By selecting a longer training time, it becomes possible to extend the prediction time beyond the training period.

Based on the findings presented in this paper, it can be concluded that the neural network methods employed in this study can be applied to similar case studies. The two case studies discussed in this paper involved a consistent number of particles throughout

the entire simulation duration, with the exclusion of the transient period from the analysis.

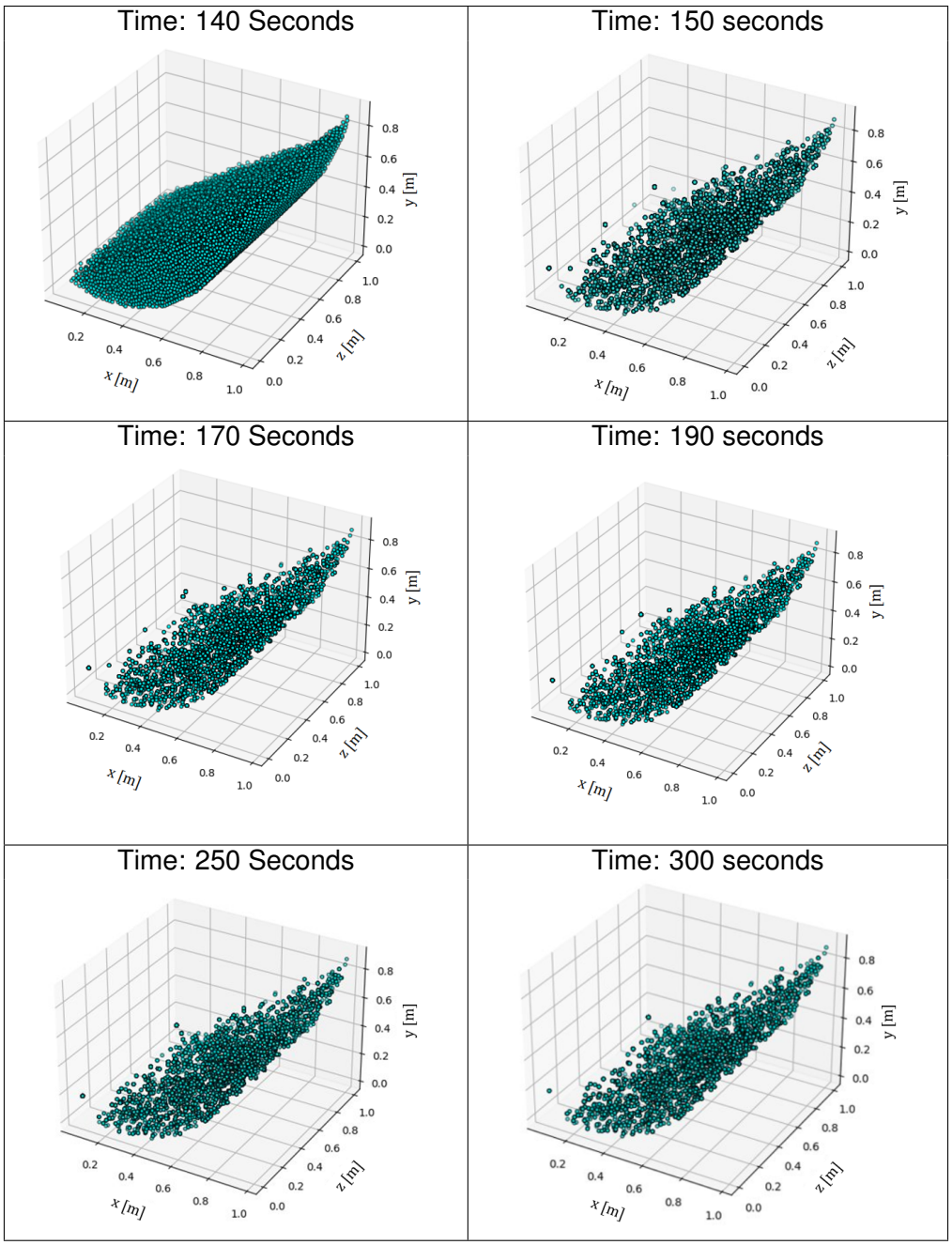


Figure 4.17: Predicting the particle's position for time beyond the training time.

4.5 Conclusion

The results obtained from the two case studies, namely the moving grate and the mixing drum, demonstrate the effectiveness of the employed neural network methods in predicting particle behavior and positions.

In the moving grate case study, the neural network's predictions for particle positions beyond the training time exhibited a reasonable pattern and were physically plausible. The average mean squared error (MSE) values for particle velocities remained consistently below a threshold of 1.4×10^{-3}, indicating the neural network's ability to accurately predict particle velocities. Additionally, the relative percent error (RPE) for particle velocities was found to be less than 5 percent, reflecting a high degree of agreement between predicted and actual values. These low error values demonstrate the accuracy of the neural network in capturing the dynamics of particle movement on the grate.

For the mixing drum case study, the neural network model successfully predicted particle positions throughout the mixing process. The average mean squared error (MSE) values for particle positions were depicted in Figure 4.15, with the maximum value observed being 1×10^{-7}. Moreover, the average relative percent error values for particle positions remained below 0.6 percent. These low error values indicate the high accuracy of the neural network in predicting particle positions during the mixing process.

The comparison of loss values between the initial and final stages of training in both case studies highlights the neural network's learning process. Initially, the network had limited familiarity with the underlying patterns of particle motion, resulting in higher loss values. However, as the training progressed, the neural network gradually acquired a better understanding of these patterns, leading to a reduction in loss values. The adoption of a recurrent neural network (RNN) and physics-informed neural network in this study proved beneficial, as it allowed the network to capture the dependencies of particle motion over time, leading to improved prediction performance.

The predicted particle positions for the mixing drum case study were visually assessed and found to exhibit a physically acceptable distribution. The particles did not exhibit

unrealistic clumping or upward movement inconsistent with gravity. These predictions remained reasonable even when the simulation was extended to a time three times longer than the training period, demonstrating the neural network's ability to generalize well and make accurate predictions beyond the training time.

Based on the obtained results, it can be concluded that the neural network methods employed in this study are effective in predicting particle behavior and positions in similar case studies. The low error values obtained for both particle velocities and positions demonstrate the accuracy and reliability of the neural network models. These findings contribute to the advancement of predictive modeling techniques for particle systems and have implications for various fields, such as industrial processes, material science, and environmental engineering.

Acknowledgement

The Doctoral Training Unit Data-driven computational modeling and Applications (DRIVEN) is funded by the Luxembourg National Research Fund under the PRIDE program (PRIDE17/12252781). https://driven.uni.lu

Chapter 5

Three-dimensional CFD-DEM simulation of raceway transport phenomena in a blast furnace[1]

[1] Navid Aminnia, Prasad Adhav, Fateme Darlik, Muhammad Mashhood, Sina Hassanzadeh Saraei, Xavier Besseron, and Bernhard Peters. “Three-dimensional CFD-DEM simulation of raceway transport phenomena in a blast furnace”. In: *Fuel* 334 (2023), p. 126574.

Abstract

Improving energy efficiency in a blast furnace (BF) has a significant effect on energy consumption and pollutant emission in a steel plant. In the BF, the blast injection creates a cavity, the so-called raceway, near the inlet. On the periphery of the raceway, a ring-type zone is formed which is associated with the highest coke combustion rate and temperatures in the raceway. Therefore, predicting the raceway size or in other words, the periphery of the ring-type zone with accuracy is important for estimating the BF's energy and coke consumption. In the present study, Computational Fluid Dynamics (CFD) is coupled to Discrete Element Method (DEM) to develop a three-dimensional (3D) model featuring a gas-solid reacting flow, to study the transport phenomena inside the raceway. The model is compared to a previously developed two-dimensional (2D) model and it is shown that the assumptions associated with a 2D model, result in an overestimation of the size of the raceway. The 3D model is then used to investigate the coke particles' combustion and heat generation and distribution in the raceway. It is shown that a higher blast flow rate is associated with a higher reaction rate and a larger raceway. A 10% increase in the inlet velocity (from 200 m/s to 220 m/s) caused the raceway volume to grow by almost 40%. The DEM model considers a radial discretization over the particle, therefore the heat and mass distributions over the particle are analyzed as well.

Keywords: coupled CFD-DEM, computational fluid dynamics, discrete element method, blast furnace, gas-solid flow, raceway

5.1 Introduction

The ironmaking industry produces 7% of the world's total carbon dioxide emissions [104]. The most frequent ironmaking process is the blast furnace (BF), accounting for more than 70% of total energy consumption in the ironmaking industry [105] and 90% of the $CO2$ emission [106]. As a result, lowering energy usage and gas pollution in the BF ironmaking operations has received a lot of attention [107].

The region inside the furnace which is created by injecting hot blast air into the coke bed is called the raceway. Its shape is affected by different parameters such as blast velocity, the geometry of the nozzles, and operational circumstances [108]. Therefore, the study of the raceway is vital to analyze the gas species distributions and heat supplies which, directly have an impact on the furnace's productivity and efficiency. As a result, the raceway has been comprehensively studied, especially the principles of the particle scale. Experimental and computational methods have been used extensively to study raceway phenomena over the past few decades. As a consequence of the severe operating circumstances within the actual BF process (e.g., high temperature and pressure situations), measurement during an experimental test is difficult to undertake and only a few studies have worked on this with very limited data, such as gas species distributions [109]. For example, Matsui et al. [110] used microwave reflection gunned via a tuyere to study the BF raceway formation under heavy coal injection rate circumstances. Sastry et al. [111] studied the particle system in a two-dimensional (2D) cold model. In another study, Sastry et al. [112] conducted experimental research in a packed bed and found that the characteristics of coke particles had a significant impact on cavity development and breakdown. Zhang et al.[113] used an image-based flame detection approach to investigate the combustion characteristics of a BF raceway and discovered that the raceway temperature profile could fluctuate considerably. These experimental studies despite helping us gain a better picture of how raceways work can only explain BF functioning at the macro-scale information such as pressure and temperature in the local spots and are not able to obtain the micro-scale information such as inter-particle/phase interactions, raceway shape/size, heat transfer, coke combustion.

In light of the limitations of experimental research and the difficulty of performing accurate measurements, numerical simulations are generally employed to study raceway phenomena. Numerical simulations are divided into two common categories: Eulerian-Eulerian and Eulerian-Lagrangian. Mondal et al. [114] studied the influences of the air blast velocity on the shape and size of the raceway zone in a BF by using the Eulerian-Eulerian model. However, the Eulerian-Eulerian model has no capacity of obtaining information such as particle

and phase interactions, particle residence time, and particle trajectory [115]. Besides the simple assumptions of inter-particle collisions in this method make it difficult to adequately capture numerous interparticle collisions near the tuyere and the quasi-static-regime in the deadman region and the associated flow features[116]. Such difficulties can be overcome by one of the important Eulerian-Lagrangian methods, the discrete element method (DEM) coupled with Computational Fluid Dynamics (CFD) named CFD-DEM. In recent years, many studies have applied CFD-DEM methods to investigate the raceway's information. For example, Xu et al. [117] proposed CFD-DEM methods to study gas fluidization on fixed and fluidized beds. To comprehend raceway formation, they illustrated that this method can capture gas-solid flow characteristics ranging from large scale (such as processing equipment) to little scale features (such as each particle). Feng et al. [118] developed a 2-dimensional model to study the particle flow in the modeling of BF, finding that both solid and gas phases flow are changed spatially and temporally, in particular in the cohesive zone, which is affected by the layered ore and coke particle structure. Yuu et al. [119] compared the characteristics of the raceway such as depth and heights with experimental data and additionally reported dynamic characteristics such as the flow of solid particles, and the airflow around the raceway. Hilton and Cleary [120] used a discrete approach and observed the effect of injection velocity and bed pressure on the formation of raceway and investigate those non-spherical particles, as opposed to spherical particles, can form the raceways at higher gas input velocities. Wang and Shen [121] developed a reacting model to study raceway formation at the particle scale and discussed the impacts of several factors on raceway combustion (such as inlet velocity, temperature, and oxygen mass fraction). To examine raceway formation, Miao et al. [122] published a 2D CFD-DEM model for full-scale BF conditions and showed that in comparison to the studies in the laboratory circumstance, the raceway parameters are substantially more complicated in full-scale BF. Cui et al. [123] used a particle scale CFD-DEM method to study the raceway cavity shape and its parameters such as heat source, mass source, and chemical reactions and additionally the effect of the gas inlet velocity, size of particles, and particle discharge rate

on the raceway formation. Dianyu et al. [124] also developed a 2D CFD-DEM model to analyze the effect of parameters such as particle size and oxygen enrichment on raceway formation and gasification rate. Recently, there has been an increasing interest in operating furnaces using renewable fuels, such as hydrogen, and many researchers have used CFD simulations to prove their efficiency[125, 126, 127]. Though this study does not consider renewable fuels, its findings can be applied to such endeavors.

The present study emphasizes the superior ability of 3D models over 2D models to predict the behavior of raceways. Therefore a 3D particle-scale CFD-DEM model of a BF raceway is developed. Using radial discretization, heat and mass transfers within particles are solved, therefore the internal gradients of particles are seen. The developed model incorporates oxidation reactions within the particles and heat and mass transfer between particles and the gas. Additionally, the impact of parameters such as inlet velocity and particle mesh on raceway size and temperature distribution is discussed. In section the governing equations of CFD and DEM models are presented along with the details of coupling techniques. In section 5.3 the results of the validation of the 3D model is presented. Then, using the comparisons between the 2D and 3D models it is argued that the inherent assumptions associated with 2D models make it unable to predict the raceway dynamics with precision. In the same section the results from the 3D model are presented and discussed. It is also shown that the discretization of the particles in the DEM model can have significant effect on the predicted size of the raceway and the gas temperature.

5.2 Model Description

XDEM software[35] is used in the current work. This software uses Lagrangian-Eulerian approach to for CFD-DEM coupling. Its multi-scale and multi-physics framework considers particles as discrete entities while fluid as a continuous medium.

Table 5.1: Nomenclature

Physical constants

α	Heat transfer coefficient ($W/(m^2.K)$)
β	Interphase momentum exchange ($kg/(m^3.s)$)
ϵ	Porosity
λ_f	Thermal conductivity ($W/(m.K)$)
μ	Dynamic viscosity ($Pa.s$)
η	Weight of particle for porosity calculation
Ω_c	Implicitly treated drag term ($1/s$)
ρ	Density (kg/m^3)

Operators

∂	Differential operator (-)
Δ	Difference (-)
∇	Nabla operator (-)

Scalars

A	Surface Area
c_p	Specific Heat ($J/kg.K$)
C_d	Drag Coefficient (-)
d	Particle diameter (m)
h	Convective heat transfer coefficient ($W/(m^2.K)$)
I_i	Moment of inertia ($kg.m^2$)
m	Mass (kg)
m'	Mass source ($kg/(m^3.s)$)
p	Pressure (Pa)
q'	Heat source (W/m^2)
q''	Heat flux (W/m^2)
r, R	Radius (m)
Re	Reynolds number (-)
t	Time (s)
T	Temperature (K)
T_{final}	Length of simulation (s)
V	Volume (m^3)

Subscripts

c	Cell
$cond$	Conduction
d	Drag
eff	Effective values
f	Fluid
i, j	Particle
n	Normal direction
p, P	Particle
s	Solid
rad	Radiation
t	Tangential direction

Superscripts

n	Geometry exponent
(n)	n^{th} (time) step
$(n+1)$	n^{th} (time) step $+1$

First order tensors (vectors)

$\vec{A_c}$	Acceleration on fluid cell due to explicitly treated drag term (m/s^2)
$\vec{g}$	Gravitational acceleration (m/s)
$\vec{F^c}$	Contact Forces (N)
$\vec{F^g}$	Gravitational Force (N)
$\vec{F^{ext}}$	External Forces (N)
$\vec{F_B}$	Buoyancy Force (N)
$\vec{F_D}$	Drag Force (N)
$\vec{M_{i,j}}$	Torque generated by inter-particle forces ($N.m$)
$\vec{v_f}$	Fluid velocity field
$\vec{X_i}$	Positional vector (m)
$\vec{\omega}$	Rotational velocity (rad/s)

5.2.1 Governing equations for discrete particles

XDEM predicts both dynamics and thermodynamics of the particulate system. The particle position, velocity and acceleration are calculated with the dynamics module of the XDEM, where as the temperature, and processes like combustion, gasification, drying etc are calculated with the conversion module of the XDEM.

Dynamics module

The Discrete Element Method (DEM) used in the dynamics module of XDEM is based on the soft sphere model. In this method, it is assumed that the particles are deformable and can overlap each other, where the magnitude of overlap is decided by the contact force using the force-displacement law. The hardness of the particle is expressed via Young's Modulus, while the particle energy dissipation is described with dampener and/or dashpot. The translational and rotational movements of individual particles are tracked using the classical mechanics equations. A detailed description of all the terms mentioned below could be found in previous work [128]. A summary of the translational and rotational motion equations is given below: Equations of particle motion:

$$m_i \frac{d\vec{v_i}}{dt} = m_i \frac{d^2\vec{X}_i}{dt^2} = \vec{F_i^c} + \vec{F_i^g} + \vec{F_i^{ext}} \tag{5.1}$$

where $\vec{F_i^{ext}}$ is the sum of all the external forces acting on the particle, such as buoyancy forces $\vec{F_B}$ (Eq 5.25) and drag forces $\vec{F_D}$ (Eq 5.26).

$$I_i \frac{d\vec{\omega_i}}{dt} = \sum_{j=1}^{n} \vec{M_{i,j}} \tag{5.2}$$

Conversion module

The particles a modelled with pores/voids. These pores are modelled to have a gaseous mixture of different chemical species. Mass conservation equation for fluid within particles

pores:

$$\frac{\partial}{\partial t}\left(\epsilon_f \rho_f\right) + \vec{\nabla} \cdot \left(\epsilon_f \rho_f \vec{v_f}\right) = m'_{s,f} \tag{5.3}$$

One dimensional transient energy conservation equations for particles:

$$\frac{\partial \rho c_p T}{\partial t} = \frac{1}{r_n}\frac{\partial}{\partial r}\left(r^n \lambda_{eff}\frac{\partial T}{\partial r}\right) - r^n\left(\vec{v}\rho_f c_{p_f} T\right) + \sum_{k=1}^{l}\dot{\omega}_k H_k \tag{5.4}$$

The mass balance and transport equation of individual fluid species within the particle pores:

$$\frac{\partial}{\partial t}\left(\epsilon_f \rho_{f,i}\right) + \nabla \cdot \left(\epsilon_f \rho_{f,i} \cdot \vec{v_f}\right) = \frac{1}{r_n}\frac{\partial}{\partial r}\left(r^n \epsilon_f D \frac{\partial \rho_{f,i}}{\partial t}\right) + m'_{s,f,i} \tag{5.5}$$

Following boundary conditions are applicable to the governing equations mentioned above:

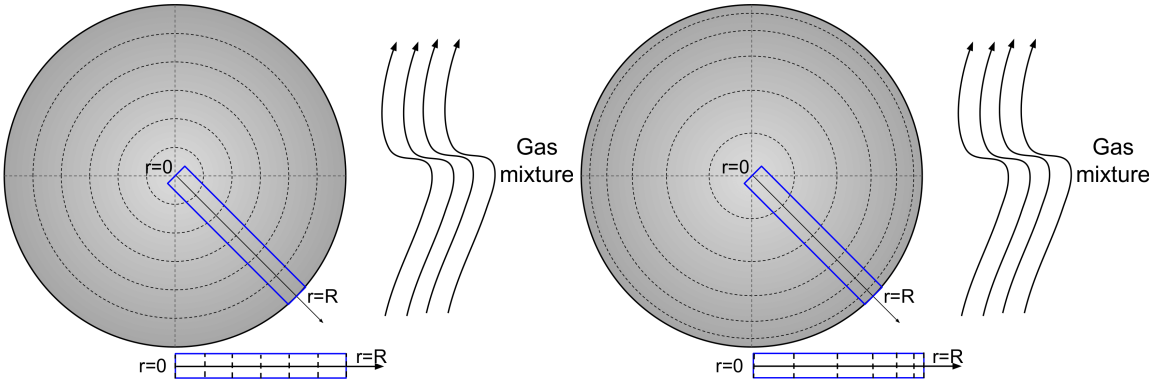
$$-\lambda_{eff}\frac{\partial T}{\partial r}\bigg|_{r=0} = 0 \tag{5.6}$$

$$-\lambda_{eff}\frac{\partial T}{\partial r}\bigg|_{r=R} = \alpha(T_R - T_\infty) + q''_{rad} + q''_{cond} \tag{5.7}$$

$$-D_{i,eff}\ \frac{\partial \rho_i}{\partial r}\bigg|_{r=R} = \beta_i\left(\rho_{i,R} - \rho_{i,\infty}\right) \tag{5.8}$$

In the Eq 5.7, q''_{cond} and q''_{rad} are conduction and radiation heat source respectively from the neighbouring particles. The detailed description of the conduction and radiation between particles is given in B. Peters [129].

In the conversion module of XDEM, a radial discretization is considered to solve for heat & mass transfer within the particle. This radial discretization can be uniform or non-uniform, as shown in fig 5.1. In the current work, non-uniform radial discretization is utilized. The non-uniform radial discretisation allows to have smaller cell length near the particle surface that allows the model to capture the sharp temperature and mass flow gradients.



(a) Uniform discretization (b) Non-Uniform discretization

Figure 5.1: Radial discretization for heat & mass transfer within a particle

5.2.2 Governing equations for fluid

In Eulerian volumetric average method, the conservation equation of mass (Eq 5.10), momentum (Eq 5.11) and energy (Eq 5.12) are written over a representative volume, where porosity (ϵ Eq 5.9) refers to the interstitial solid space particles. These governing equations for fluids are given below. Detailed description of the porosity calculation can be found in [128], the porosity calculation in brief is as follows, where V_c is CFD cell volume, V_i is the particle volume of i^{th} particle in the CFD cell and η is weight for porosity calculation:

$$\epsilon = 1 - \frac{1}{V_c}\sum_{i}^{n} \eta_i V_i \tag{5.9}$$

Conservation of mass

$$\frac{\partial}{\partial t}\left(\epsilon\rho_f\right) + \nabla\cdot\left(\epsilon\rho_f\vec{v}_f\right) = m' \tag{5.10}$$

Conservation of momentum

$$\frac{\partial}{\partial t}\left(\epsilon\rho_f\vec{v}_f\right) + \nabla\cdot\left(\epsilon\rho_f\vec{v}_f\vec{v}_f\right) = -\epsilon\nabla p + \epsilon\rho_f\vec{g} + \epsilon\rho_f\vec{A_c} + \epsilon\mu\nabla^2\vec{v}_f - \epsilon\rho_f\Omega_c\vec{v}_{f_c} \tag{5.11}$$

Conservation of energy

$$\frac{\partial}{\partial t}\left(\epsilon\rho_f h_f\right) + \nabla\cdot\left(\epsilon\rho_f\vec{v}_f h_f\right) = \frac{\partial p}{\partial t} + \epsilon\vec{v}_f\cdot\nabla p + q' \tag{5.12}$$

Chemical reactions are also considered in the CFD solver. In a multispecies gas mixture, the mass conservation equation for a species i, is given in Eq 5.13:

$$\frac{\partial}{\partial t}\epsilon\rho_{f,i} + \nabla\cdot\left(\epsilon\rho_{f,i}\cdot\vec{v}_f\right) = m_i' \tag{5.13}$$

5.2.3 Chemical reactions

The current study focuses on the raceway and areas immediately next to raceway. In this region mainly gasification and combustion reactions are observed in the particles, the (solid phase) reactions are presented in Eq 5.14, 5.15, and 5.16. Considering the temperatures in and near raceway, the reaction 5.14 producing CO is mainly observed. Some small amount CO_2 is produced as shown in reaction 5.15, but due to high temperatures ($\geq$ 1073 K [130]) it quickly decomposes to CO as shown in gas phase reaction 5.16.
The gasification reactions are as follows:

$$C + 0.5\,O_2 \rightarrow CO \tag{5.14}$$

$$C + O_2 \rightarrow CO_2 \tag{5.15}$$

$$C + CO_2 \rightarrow 2\,CO \quad (Boudard's reaction) \tag{5.16}$$

As opposed to the previous reactions, reaction 5.17 is taking place in purely gaseous state (handled by CFD solver). Due to the high temperatures in the region of interest, it is observed that the CO_2 produced from the following chemical reactions, converts back to

CO according to reaction 5.16.

$$CO + 0.5\,O_2 \rightarrow CO_2 \tag{5.17}$$

Generally reactions can be written as follows:

$$\sum_{i=1}^{N} \nu_i' \, R_i \;\rightleftharpoons\; \sum_{j=1}^{M} \nu_j'' \, P_j \tag{5.18}$$

where N denotes the number of reactants R_i, M denotes the number of products P_j and $\nu_{i/j}$ represents the absolute values of the corresponding stoichiometric coefficient.

$$\dot{\omega} = -\frac{1}{\nu_i'}\frac{dc_i}{dt} = \frac{1}{\nu_j''}\frac{dc_j}{dt} \tag{5.19}$$

The actual reaction rate $\dot{\omega}$ may depend on species concentrations, the available reactive surface O_{sp} and the temperature; so that in general $\dot{\omega} = f(c_i,\, c_j, O_{sp},\, T,\, \ldots)$. Thus, an Arrhenius law is employed to describe the temperature dependency of the reaction rate as

$$k(T) = k_0 \, e^{\left(\frac{-E_a}{RT}\right)} \tag{5.20}$$

where $k(T)$ represents the temperature dependent rate coefficient, k_0 referred to as frequency factor and E_a denotes the activation energy.
If thermodynamic equilibrium is reached, then an equilibrium constant $K_{eq,c}$, representing the thermodynamically equilibrium state, can be obtained as

$$K_{eq,c}(T) = \frac{k_f(T)}{k_b(T)} = \frac{\prod_{j=1}^{M} c_{eq,P_j}^{\nu_j''}}{\prod_{i=1}^{N} c_{eq,R_i}^{\nu_i'}} \tag{5.21}$$

In the XDEM software, the equilibrium constant $K_{eq,c}(T)$ is calculated as

$$K_{eq,c}(T) = e^{\frac{A_{eq}}{T} + B_{eq}} \tag{5.22}$$

where A_{eq} and B_{eq} are constant values that may come from existing tables or from equilibrium diagrams of phase diagrams.

Table 5.2: Chemical reaction rates

Variable	Reaction 5.14	Reaction 5.17
E_a	$149,000$	$20,129$
A_{eq}	0	$2.24e+08$
B_{eq}	0	0
Temperature Range	$273K$ to $1500K$	

5.2.4 CFD-DEM Coupling

The CFD-DEM coupling is achieved through conventional staggered approach. In this approach, the output from one simulation (solver) is used as an input for the other. Considering current work, assume that solver S_1 is the CFD solver, and the solver S_2 is DEM solver. The fluid solver S_1 solves the momentum, mass, reactions and energy equations for the fluid. The fluid solver output such as the fluid velocity, temperature, species mass fraction etc., are then used as boundary conditions for the particles in DEM solver S_2. The DEM solver S_2 uses solution from CFD solver, to compute various source terms by computing the momentum and energy equations for particles. In the next time step, these source terms are communicated to the fluid solver S_1, which then uses the solution from n^{th} time step to get a new solution for the $(n+1)$ time step.

$$x_2^{(n+1)} = S_1^{(n)}\left(x_1^{(n)}\right) \tag{5.23}$$

In Eq 5.23, CFD solver S_1 uses old time step's boundary value (or in case of first time step it can be the boundary conditions or an initial guess), $x_1^{(n)}$to compute the values of x_2 for next time step $x_2^{(n+1)}$. During this time, DEM solver S_2 waits for CFD solver S_1 to compute solution and exchange the updated solution $x_2^{(n+1)}$.

$$x_1^{(n+1)} = S_2^{(n)}\left(x_2^{(n+1)}\right) \tag{5.24}$$

In Eq 5.24, the updated solution x_2 is used to update the solution for x_1 for the next time step. This can also be seen graphically in Fig 5.2.

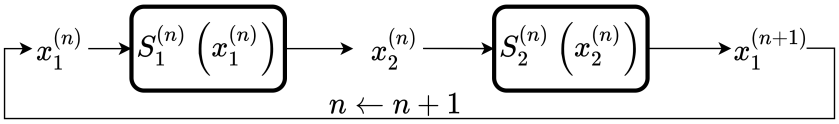


Figure 5.2: Serial Staggered Coupling Scheme

Fluid forces on particles

There are two types of fluid forces acting on the particles, namely hydrostatic force and hydrodynamic force. The hydrostatic force is the buoyancy force which accounts for the pressure gradient around an individual particle [131].

$$\vec{F_B} = -V_{p_i}\nabla p \tag{5.25}$$

In the Eulerian-Lagrangian approach, the hydrodynamic force corresponds to the fluid-particle interaction. This force depends on the relative velocity of the solid particle and fluid along with the forces acting due to presence of neighbouring particles. The drag force acting on the particle due the fluid for CFD-DEM approach is given as follows:

$$\vec{F_D} = \frac{\beta V_p}{(1-\epsilon)}\left(\vec{v}_f - \vec{v}_p\right) \tag{5.26}$$

The interphase momentum exchange β is predicted according to Gidsaw [132]. Although to cover all range of void fraction (ϵ), Wen and Yu [133] ($\epsilon \geq 0.8$) and Ergun and Orning [134] ($\epsilon < 0.8$) equations are included.

$$\beta = \begin{cases} 150\dfrac{(1-\epsilon)^2}{\epsilon}\dfrac{\mu_f}{d_p^2} + 1.75(1-\epsilon)\dfrac{\rho_f}{d_p}|\vec{v}_f - \vec{v}_p|, & \text{if} \quad \epsilon < 0.8 \\ \dfrac{3}{2}C_d\dfrac{\epsilon(1-\epsilon)^2}{d_p}\rho_f|\vec{v}_f - \vec{v}_p|\epsilon^{-2.65}, & \text{if} \quad \epsilon \geq 0.8 \end{cases} \tag{5.27}$$

where the drag coefficient C_d is given as:

$$C_d = \begin{cases} \dfrac{24}{Re}\left[1 + 0.15(Re)^{0.687}\right], & \text{if} \quad \text{Re} < 1000 \\ 0.44, & \text{if} \quad \text{Re} \geq 1000 \end{cases} \tag{5.28}$$

and the Reynolds number for the particle is given as:

$$Re = \frac{\epsilon\rho_f|\vec{v}_f - \vec{v}_p|d_p}{\mu_f} \tag{5.29}$$

Particle momentum source terms

In the fluid, the drag exerted by the solid particles is treated in semi-implicit way according to the method proposed by Xiao and Sun[135]. The explicit momentum source term $\vec{A}_c$ and implicit momentum source term Ω_c are as given in Eq 5.30

$$\vec{A}_c = \frac{1}{\rho_f V_c}\widetilde{\sum}_i B_i \vec{u_{p_i}}, \quad \Omega_c = \frac{1}{\rho_f V_c}\sum_{i=1}^{c_n} B_i \tag{5.30}$$

Particle heat and mass source terms

Fluid flow conditions such as fluid temperature, specific heat, thermal conductivity, species mass fractions are exchanged from CFD to DEM. These are used as boundary conditions for solving energy balance, mass balance and reaction equations for particles.

Based on the energy balance equations, heat loss/gain due convection or due to change

in composition of particles is computed. This is used as the (explicit) heat source in fluid energy equation. Similarly, mass source and species mass fraction source are computed.

$$q_i' = h_i A_i (T_{p_i} - T_f) \tag{5.31}$$

where h_i is the heat transfer coefficient for a given particle i, which is a function of Re, Pr, λ_f, d_p and cell porosity.

As there are different phenomenon driving mass transfer, such as evaporation, mass flux due to the gradient of species concentration, species production due to chemical reactions, a generalised way to represent individual species mass source is as follows:

$$m'_{s,f,i} = (Areaof MassTransfer) \times (MassTransferCoefficient) \times (DrivingForce) \tag{5.32}$$

The total mass transfer is summation of all the species mass transfer terms.

$$m' = \sum_i m'_{s,f,i} \tag{5.33}$$

5.2.5 Computational Procedure

A schematic for the CFD-DEM coupling is shown in fig 5.3. For the , XDEM and OpenFOAM libraries are linked together as a single executable. The simulations starts after running the executable. In the first step, DEM is initialized, where all the particles, walls, domain and XDEM mesh are created. After creating the mentioned assets, initial boundary conditions for the particles are applied. An initial porosity (ϵ) field is computed. Next CFD is initialized, where geometry and mesh is generated, boundary conditions on the fluid domain are applied, as well as porosity is made available to CFD solver, so that CFD solver takes into account the presence of particles from the first time step itself. But, these particles do not contribute to any heat or mass source terms of the fluid governing equations at the first time step. After all the required initializations, the time loop starts. The fluid governing equations are solved by the CFD solver developed with the assistance of OpenFOAM. The

data transfer between DEM and CFD is done via direct read/write in memory. All the data is stored on the OpenFOAM mesh. After exchanging the data from CFD to DEM, the CFD data is used as boundary conditions for the particles. XDEM then solves the governing equations for the particles, and writes output fields such as porosity, momentum, heat, mass and species mass fraction source. After writing the data, the time loop proceeds to next time step ($T + dT$). In this time step, the data written by XDEM i.e various source terms, are injected in fluid governing equations. In this way the simulation continues until the specified end time T_{final}.

5.2.6 Simulation setup

For the simulations presented, the region of interest is only the raceway, and not the whole BF. The raceway is modelled as a 3D box of dimension $0.6m \times 0.6m \times 1.5m$. For the 2D case the cross-section dimensions remain the same ($0.6m \times 1.5m$).

The boundary conditions for the CFD and DEM are described in the Table 5.3 and 5.4 respectively. It is assumed that the particles are preheated to 1300 K and the inlet air enters at a temperature of 1500 K. Since the primary purpose of this research is to demonstrate the effects of 3D simulations and particle discretization, the particles are spherical and of the same size to eliminate particle shape and size effects.

5.3 Results and Discussion

5.3.1 Model validation

As it was explained in section 5.2, the CFD-DEM model used in this study is developed by coupling a CFD model in OpenFOAM and a particle system model in XDEM. The validations of the coupling have been presented in previous studies [35, 136]. For verifying the particle-scale reaction models and gas-solid reactive interactions, the resulting gas composition from the 3D model is validated against an experimental hot model[137].

Table 5.3: Simulation conditions for CFD

Variable	Value
2D grid	20×50 1000 Hex elements
3D grid	$20 \times 20 \times 50$ $20,000$ Hex elements
Inlet Specie Mass fraction	
CO	$0.00\,[-]$
$CO2$	$0.00\,[-]$
$N2$	$0.79\,[-]$
$O2$	$0.21\,[-]$
Specie Mass fraction inside simulation domain	
CO	$0.0\,[-]$
$CO2$	$0.0\,[-]$
$N2$	$0.79\,[-]$
$O2$	$0.21\,[-]$
Time step length	$0.005\,s$
Simulated Time	$20.0\,s$
Temperature	
Inlet	$1500\,K$
Internal Domain	$1500\,K$
Turbulence Model	$k-\epsilon$ Reynold's Averaged Simulation (RAS)

Table 5.4: Simulation conditions for DEM

Variable	Value
Particle Shape	Spherical
Particle Diameter	$0.01\,m$
Particle Density	$1111.1\,kg/m^3$
Particle Initial Temperature	$1300K$
Particle Composition	
Char (Solid)	$0.97\,[-]$
Light Ash (Solid)	$0.03\,[-]$
Number of particles	$10,000$
Particle radial discretization	5 uniform cells
Time step length	$0.005\,s$
Simulated Time	$20.0\,s$
Mechanical Properties	
Contact Model	Hertz Mindlin
Spring Constant	$100.0\,kN/m$
Viscous Contact Damping	$2420000.0\,N.s/m$
Friction Coefficient	$0.8\,[-]$
Poisson's Ratio	$0.45\,[-]$
Young's Modulus	$500000.0\,Pa/MPa$
Thermal Properties	
Thermal Conductivity	$0.47\,W/m.K$
Specific Heat	$1500\,J/kg.K$
Molar Mass	$30\,[-]$

Figure 5.5 shows the comparison of the simulation results for oxygen, nitrogen, and carbon monoxide against the experimental measurements. The results show that the model can predict the trend in coke combustion (oxygen consumption and carbon monoxide production) to an acceptable degree. However, there is a deviation between the predicted and measured values for oxygen and carbon monoxide which suggests and underestimated reaction rate for the coke combustion. The same level of deviation was also observed in previous studies [137, 124]. This deviation is rendered as acceptable considering the harsh measurement conditions inside the furnace and thus the uncertainty associated with the measurements.

5.3.2 Comparison of 2D and 3D simulations of raceway

First, we propose to study the differences between 2D and 3D simulations of the raceway. The 2D model is presented in a previous work [136]. As mentioned before, there have already been a lot of efforts in developing 2D models of the raceway and the BF. Simulations in 2D have the advantage of lower computational cost, but they come at the expense of numerical accuracy. Undoubtedly there should be a reasonable trade-off between the advantages and disadvantages. In the present work, 2D and 3D simulations are compared and the results reveal a significant discrepancy in raceway behavior. In order to make a valid comparison, the 2D and 3D cases were similar in size and mesh in the x and z direction. There are also similar initial conditions, including packed bed height and particle size.

Figure 5.6 shows the comparison of the raceway cavity in 2D and 3D simulations of the dynamics of a BF raceway. It can be observed that in 2D, the raceway cavity has larger dimensions compared to the 3D results. The height and width of the raceway in the 2D case are respectively 100 cm and 38 cm, while in 3D they are respectively 30 cm and 12 cm. This notable discrepancy between the 2D and the 3D simulations can be explained by the fact that in 3D, the momentum of the inlet flow is partly consumed to expand the raceway in the third direction, depth (which reaches up to 24 cm). Whereas in 2D simulation the momentum of the inlet air is wholly saved to expanding the raceway height and width, thus resulting in an unjustifiably large raceway. The result is that in the 2D simulation, the packed bed is expanded to fill the whole available domain whereas in reality the top of the packed bed is just raised a fraction of the packed bed height. This behavior can be observed in the previous study as well [124]. This phenomenon can be confronted by defining different initial or boundary conditions to constrain the packed bed height or fill the whole domain from the beginning but nevertheless, it would not change the fact that the dynamics of the packed bed and the raceway are misrepresented.

Moreover, the implicit assumption made for the 2D model by itself leads to a significant gap between the model and actual physics. The 2D simulation of the raceway assumes a symmetrical placement of the raceway in the BF. For 2D simulation to represent the real

BF raceway, either the BF should be a thin slice with two raceways on the opposite sides, which is naturally far from the reality, where we have a cylindrical furnace with multiple injection inlets located on the periphery; or, the inlet of the raceway should be an open slit covering the periphery which would lead to a torus raceway in the whole furnace. None of these two cases mimick the actual physical geometry to a good approximation. Due to such observations, the previous studies have noted [116] that there should be special boundary conditions defined on the domain for the 2D or quasi-3D model to represent the physics better.

However, in a 3D simulation, the actual physics of the problem is represented more accurately. As is presented in the following sections, the raceway enlargement is more confined and the packed bed movement is very limited compared to a 2D simulation. Therefore, the significant difference between the two cases led to the conclusion that 3D simulations, despite their computational costs, provide a much more reliable insight into the physics of the raceway. Additionally, with the increasing trend in computational power and thanks to parallelization, 3D simulations are becoming more affordable.

5.3.3 Typical transport phenomena of the raceway

Figure 5.7 provides a series of snapshots from the 3D simulation of a raceway section in an operating BF. The pictures depict the raceway formation as a hot air blast is injected horizontally into the furnace via the tuyeres. Following the blast, the raceway forms in the early time steps. It first penetrates the packed bed to the maximum possible depth in a horizontal direction which is approximately 24 cm, then adopts an upward anti-clockwise direction to develop further in height and eventually reach a maximum height of 33 cm from the bottom of the furnace. As can be seen in the figures, by the time of 20 s the raceway has already adopted a respectively steady shape and dimensions. Figure 5.8 shows the penetration profiles of the raceway through time in X-direction (depth), Z-direction (height) and Y-direction (width). This plot supports the observations in figure 5.7, by showing that the raceway dimensions achieve stability in all directions by 20 s, despite abrupt fluctuations

in the beginning. However, the width of the raceway shows less stability because it is comparatively confined by the walls.

As a more clear depiction of the 3D case, figure 5.9 shows the location of the raceway and the gas flow streamlines that start from the inlet and spread all through the raceway. As can be seen in the figure, the gas flow has a high velocity inside the raceway (more than 50m/s) and as it penetrates into the packed bed, loses its momentum and its velocity decreases drastically.

5.3.4 Heat and mass distribution in the packed bed

Figure demonstrates the particles and their respective temperature at three different time steps on horizontal and vertical slices. These horizontal and vertical slices are cut between two XY and XZ planes respectively located at Z=0.15 cm, Z=0.21 cm, and Y=0.27 cm, Y=0.33 cm. As can be seen in the XZ slices (subfigures a-c), the packed bed has increased in height because of the minor fluidization caused by the flow inside the raceway. The packed bed's top surface shows a downward slope from right to left, representing a general anti-clockwise flow flowing through the packed bed.

The particles preheated to an inital temperature of 1300 K. Figure shows that as the raceway forms, the temperature of particles around the ring-type zone rises. The ring-type zone is the boundary of the raceway and the location where the incoming air meets the coke particles. The oxygen concentration in this zone is high, causing a high rate of the exothermic oxidation reaction with coke. Heat is produced at a faster rate when the reaction rate is higher, therefore the temperature is raised faster in the areas close to the ring-type zone.

The distribution of O_2 and CO at t=0.1 s , t=1 s and t=20 s can be seen in figure . In the initial time step (t=0 s) the mass fraction of both O_2 and CO is zero and only Nitrogen (N_2) is present(which is not shown here for the sake of brevity). As the simulation process starts, O_2 is blasted into the furnace and CO is produced. It can be observed that the raceway region is the source of O_2 and distributes it around the packed bed. This explains the correspondence

of the O_2 distribution with the raceway shape. However this correspondence is very rough because some O_2 distributs into the packed bed via diffusion and convection and therefore go beyond the raceway perimeter. The O_2 concentration is maximum in the raceway and zero beyond the ring-type zone because the coke particles in the zone consume the O_2 and produce CO. Consequently, due to the outward flow direction around the raceway, CO transports away from the raceway. As a result, CO concentrations inside the raceway are lower than outside.

5.3.5 Influence of blast flow rate

Inlet velocity is a physical parameter that can be manipulated to achieve the desired outcome in the raceway. It directly affects the size of the raceway and thus the distribution of temperature and gas species. In this study, three test cases with inlet velocities of 180 m/s, 200 m/s, and 220 m/s are used to examine the effects of inlet velocity on raceway transport phenomena.

In all the cases, the initial inlet velocity is 10 m/s and it increases linearly with time, up to the desired inlet velocity (180, 200, or 220 m/s) at t=0.5 s. This gradual velocity increase was done both for the sake of stability and imitating the actual process. As can be seen in figure , increasing the inlet velocity results in a larger raceway, thus pushing the ring-type zone outward. As the raceway approaches proximate stability, the volumes of the raceway cavity for 180 m/s, 200 m/s, and 220 m/s are respectively 4.24, 4.72 and 6.56 cubic decimeters. These values are calculated based on considering the raceway as the region with a porosity equal to or larger than 0.7.

Figure b shows the variation of temperature along the horizontal line starting from the tuyere tip. In all three cases, somewhere in the middle of the horizontal line, there is a temperature peak. The high combustion rate of the particles and the resulting heat causes the temperature to rise. Based on the raceway profiles in figure a these peaks occur in the ring-type zone which is located just outside the raceway perimeter. The same behavior can be observed in O_2 and CO concentration profiles in figure . As is expected,

at the ring-type zone (for instance, between 0.2 m- 0.4 m for 180m/s case), there is a rapid consumption of oxygen, concurrent with the formation of carbon monoxide, which reaches its maximum concentration at the end of the probe line. This consistency of the temperature and concentration profiles with the raceway profile can be observed for all three inlet velocities. However, as the blast inlet velocity is increased, the ring-type zone is pushed further outward therefore the temperature peak and correspondingly the concentration profiles' inclination, occur at a further distance from the tuyere tip.

5.3.6 Heat and mass distribution within the particles

As described in section , the DEM model used in this study considers discretized particles. Therefore the heat and mass distributions inside the particles are considered, featuring the internal gradients of temeprature and species concentration within the particles.

Figure compares the results between a case with 1-cell particles and a case with 5-cell particles to demonstrate the significance of particle discretization. The primary difference between a 1-cell and a 5-cell particle is that in the 5-cell case there is a gradient of species and temperature within the particle. Coke combustion is driven by oxygen which diffuses into the particle from the ambient gas. Therefore the oxygen concentration has a profile within the particle, decreasing from the surface to the center. The available oxygen concentration determines the rate of coke combustion. Therefore the combustion reaction rate will have a negative gradient from the surface to the center, resulting in more heat generation (due to the exothermic reaction of coke combustion) in the cells near to the surface. Since the particle surface temperature is higher in the 5-cell case, there is a stronger heat convection with the surrounding ambient gas and the gas is thus hotter. This anticipation is clearly demonstrated by figure 5.13(b). The case with 5-cell particles shows a higher gas temperature peak in the ring-type zone. This higher gas temperature creates a higher pressure inside the raceway which pushes the raceway ceiling upward (as there is less resistance to vertical expansion compared to horizontal expansion which is limited by the right wall). This explains the larger raceway cavity for 5-cell particles as illustrated in figure (a).

However, it is important to note that although in the 5-cell case there is a gradient in reaction rate through the cell, the average reaction rate, or in other words the coke and oxygen consumption is almost the same in the two cases. Figures 5.13 (c) and (d) show respectively CO and O_2 concentration along the horizontal line from the tuyere tip. It can be observed that there is almost no difference between the two cases in O_2 consumption and CO production.

The aforementioned gradient of temperature can be significant in some particles that are subject to higher O_2 concentration. Figure shows the temperature distribution over the particle radius for two different particles located at different points in the packed bed. Both of these two cases include particles with 20 cells so that this gradient is expressed more clearly. One is inside the ring-type zone, undergoing higher reaction rates and higher temperatures. The other particle, particularly in the second half of simulation, experiences lower temperatures, suggesting that it evaded being trapped in the ring-type zone and maintained a position where heat loss and heat gain by the particle are in equilibrium.

The figures demonstrate the particle discretization which is non-uniform as it was explained in section . The cell adjacent to the particle surface adopts a minimum size and towards the center of the particle the cell size increases according to a geometric progression. It can be observed that in both cases there is a high gradient of temperature near the surface and this gradient increases over time, because of the low conductivity of coke. This sharp gradient would not be captured in a particle with no discretization [121, 123]. Therefore such models in which the whole particle is considered as a single element with a uniform temperature all over it, incorporate a rough temperature in the Arrhenius model and thus underestimate the reaction rates at the surface of the particle. As demonstrated in previous sections, in an application such as a BF where there is a complex interdependency between the heat transfer, reactions and the dynamics of the system, such gaps in the model will introduce noticeable and unacceptable errors.

5.4 Conclusion

A CFD-DEM model was developed to analyze the raceway transport phenomena in an iron-making blast furnace. The study proposed the significance of simulating the raceway in 3D. Based on the provided results, it was argued that the 2D model, due to the implicit assumptions associated with it, overestimates the raceway size. Therefore the 3D model was used to simulate the raceway and analyze the dynamical evolution of the raceway, combustion of the coke particles, and heat and species distribution in the gas flow.

It was shown that the coke combustion rate is the highest in the periphery of the raceway, known as the ring-type zone. Therefore almost all of the incoming oxygen is consumed near the ring-type zone and the particles in this region experience the highest temperatures. The discretization of the particles made it possible to analyze the heat distribution within the particles. It was shown that for the particles exposed to high oxygen concentration and high combustion rate, the surface of the particle is subjected to a relatively high temperature gradient. Therefore, in a blast furnace where thermal conductivity of coke particles is low but temperature levels are high, using a DEM model with discretized particles is an effective strategy for preventing the underestimation of particle temperatures.

The influence of the gas inlet velocity was also investigated. It was shown that higher inlet velocity results in larger raceway cavities and more penetration into the path of the packed bed. However, the temperature ranges of the raceway and reaction rates do not follow a clear correlation with the inlet blast flow rate. These findings offer insight into the complex correlations between the dynamics and thermodynamics of the raceway.

Acknowledgment

This research was partially supported by Luxembourg National Research Fund (project numbers 13558062 and 14843353).

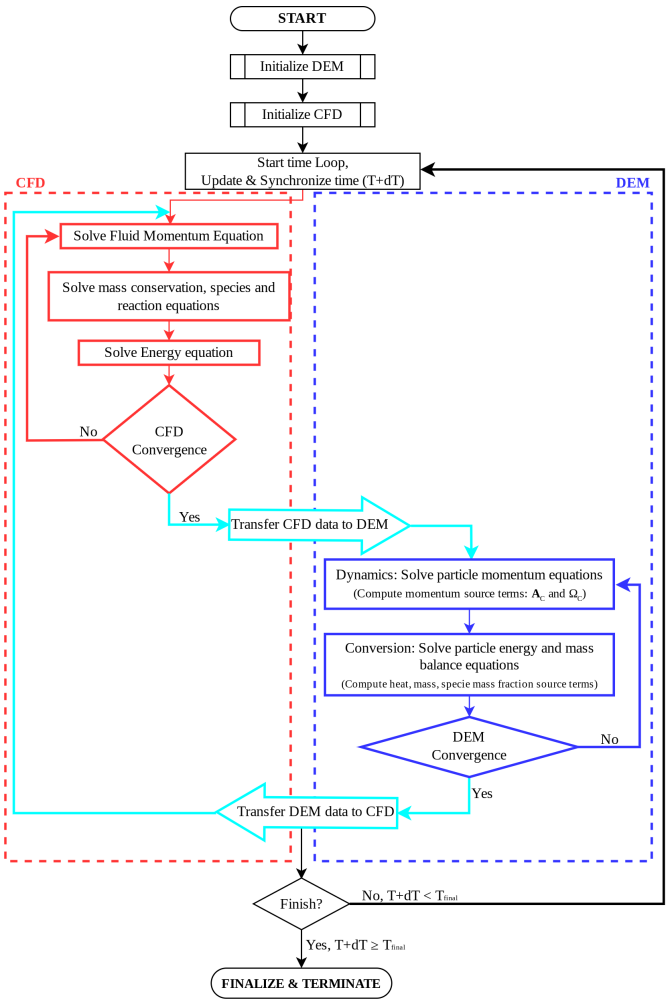


Figure 5.3: Flow chart of coupled OpenFOAM-XDEM solver showing calculation steps and exchange of data

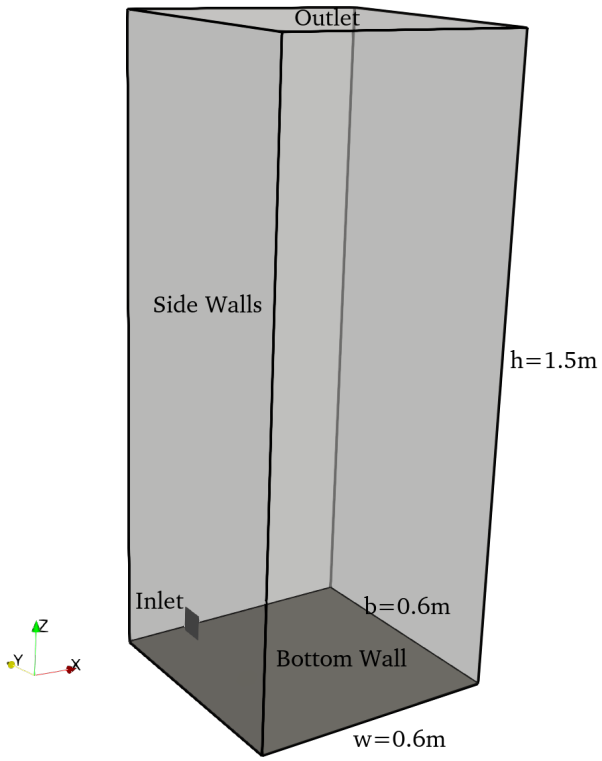


Figure 5.4: Simulation geometry and Boundaries

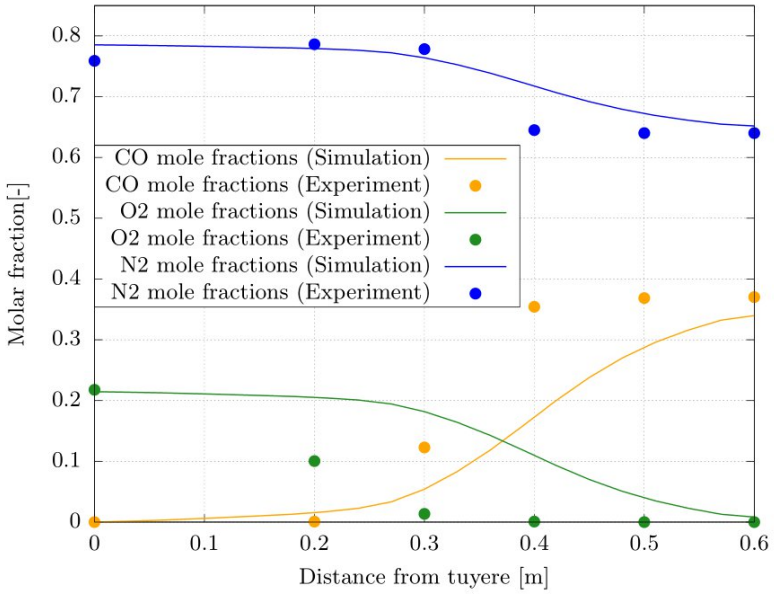


Figure 5.5: comparison of the measured and predicted gas compositions along the central axis of the tuyere

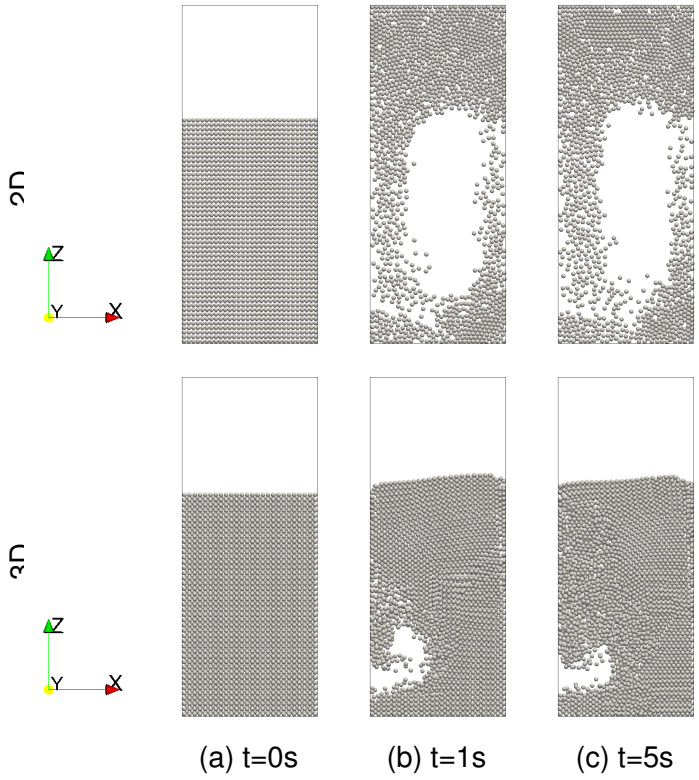


Figure 5.6: The 2D and 3D dynamics simulation of a blast furnase raceway at three different time steps. The 3D results are slices from the 3D packed bed cut between two XZ planes on the two sides of the inlet

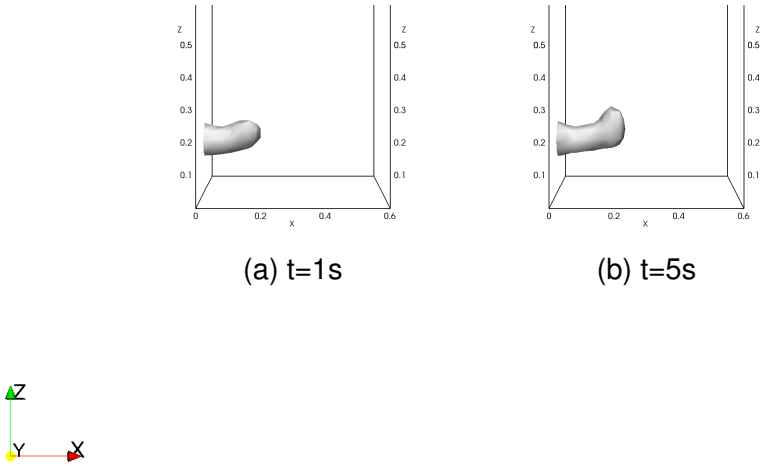


Figure 5.7: Topological evolution of the raceway cavity at different time steps in the 3D simulation of a BF raceway with a blast velocity of 200 m/s

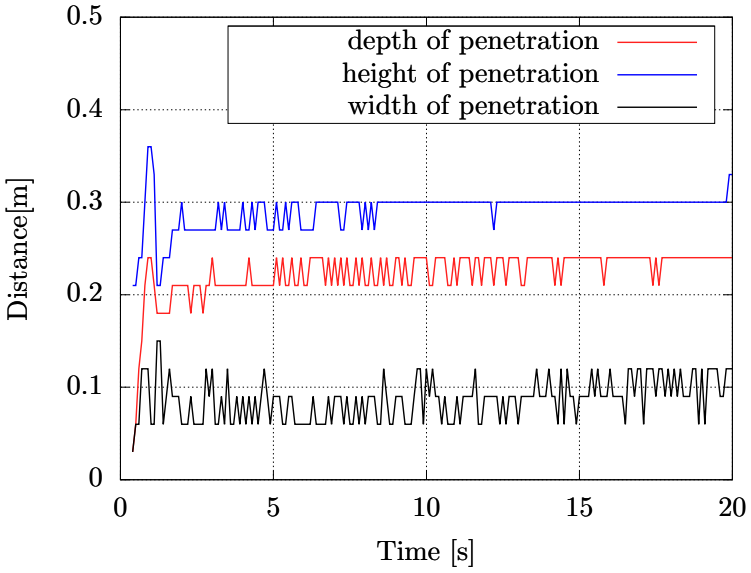


Figure 5.8: Temporal variation in depth (starting from tuyere tip in the direction of inlet flow, i.e. inlet central line) , height (starting from the bottom of the furnace in the positive z-direction) and width (starting from inlet center line in the positive y-direction) of the raceway cavity calculated based on the porosity isoline of 0.7

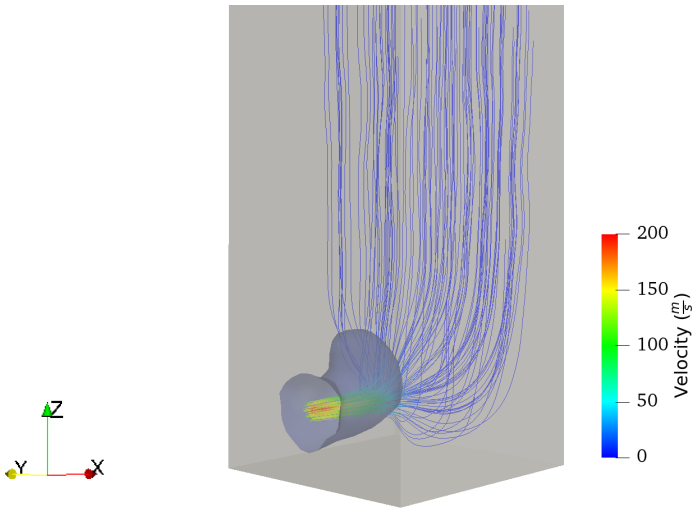


Figure 5.9: Streamlines of the flow passing through raceway for the 3D case with an inlet velocity of 200 m/s. The streamlines are colored by the flow velocity.

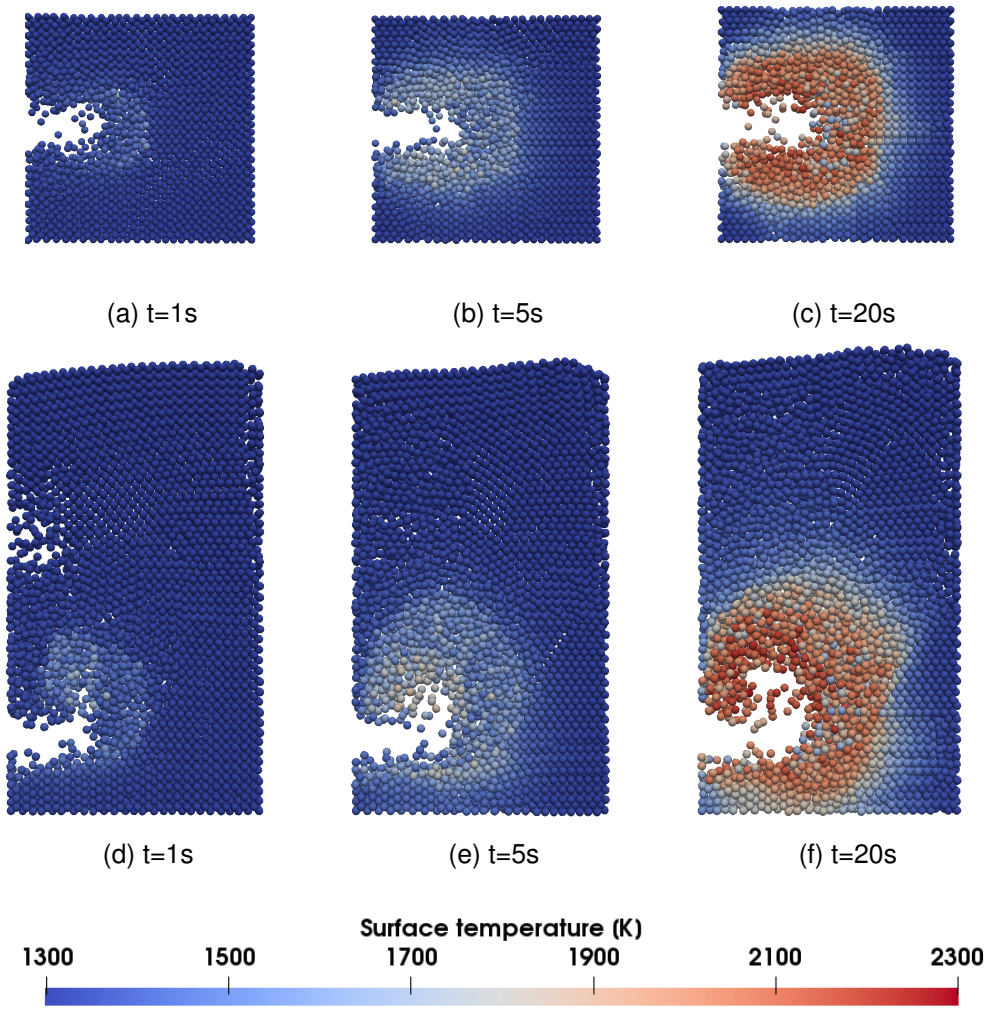
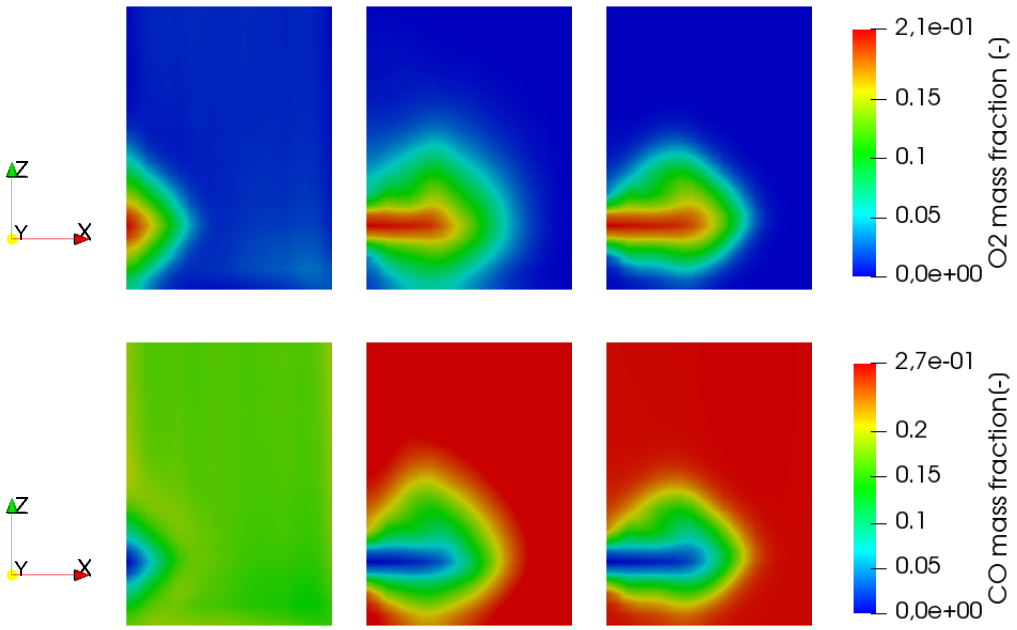
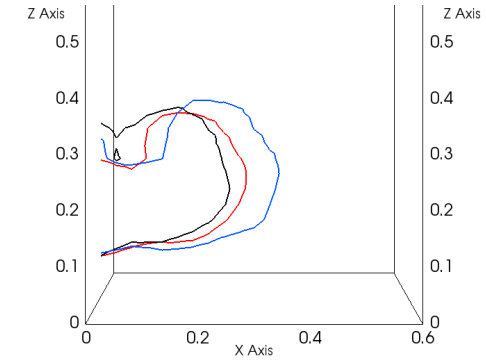


Figure 5.10: Spatial distributions of coke particles in the raceway packed bed at different time steps in a horizontal slice cut from the 3D packed bad, between two XY planes on the opposite sides of the inlet (a,b,c) and a vertical cut between two XZ planes on the opposite sides of the inlet (d,e,f). Particles are colored by the surface temperature of the particles.

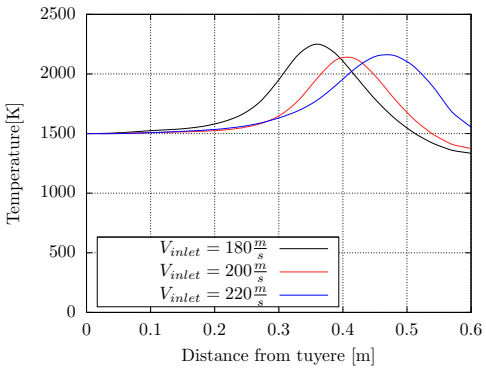


(a) t=0.1s (b) t=1s (c) t=20s

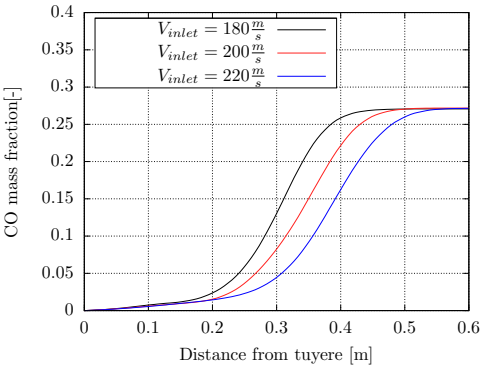
Figure 5.11: O_2 and CO mass fraction distribution in the gas (CFD) at different time steps of the 3D simulation on an XZ slice located on Y=0.3 m which passes the inlet



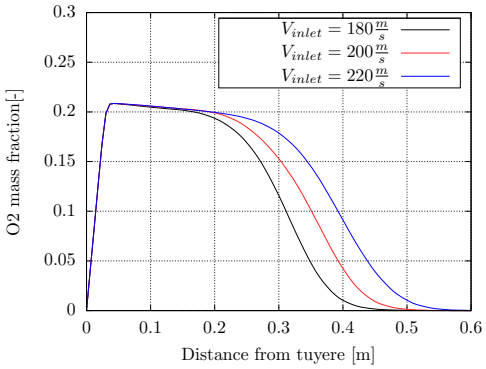
(a) Raceway profiles



(b) Gas Temperature profiles

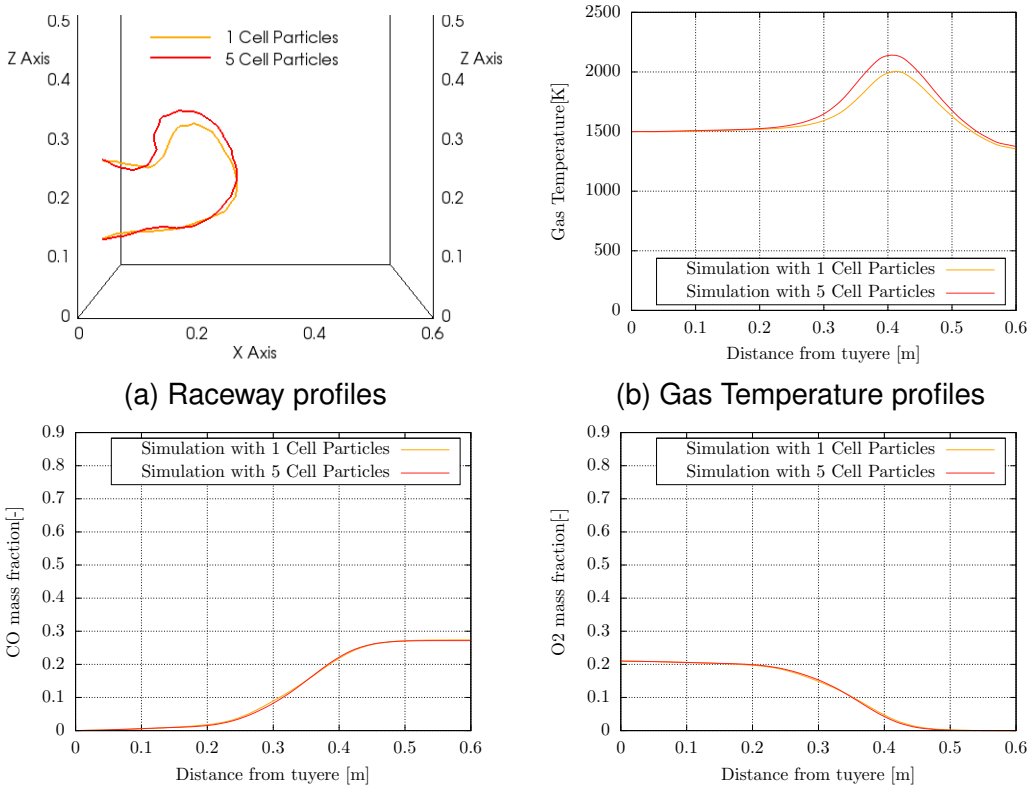


(c) CO concentration profiles



(d) O_2 concentration profiles

Figure 5.12: Comparison of racewy profile, temperature, CO concentration and O_2 concentration along the horizontal line from the tuyere tip for three differenet velocities 180 m/s (black), 200 m/s (red) and 220 m/s (blue)



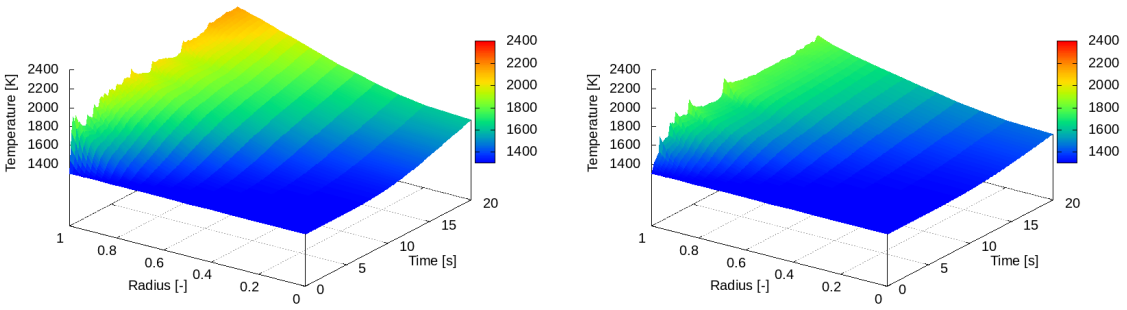
(a) Raceway profiles

(b) Gas Temperature profiles

(c) CO concentration profiles

(d) O_2 concentration profiles

Figure 5.13: Comparison of racewy profile, gas temperature, gas CO concentration and gas O_2 concentration along the horizontal line from the tuyere tip for two cases: particles with one internal cell and thus no discretization(Orange) and particles with 5 cells (Red)



(a) inside the ring-type zone

(b) off the ring-type zone

Figure 5.14: Temporal and spatial distribution of temperature in two coke particles:(a) one, positioned inside the ring-type zone and (b) the other, further away from the ring-type zone

Chapter 6

Conclusion

6.1 Conclusion

Physics-Informed neural networks (PINNs) have emerged as a promising approach to accurately represent the complex motion behavior of granular materials.

The motion of granular flow in the fluid environment is governed by physical factors, including interaction forces. The Navier-Stokes equation, which describes the motion of fluid, plays a crucial role in understanding the fluid flow around granular materials.

By combining principles from physics and deep learning, PINNs leverage their ability to capture the patterns in predicting the particle's motion. These networks can effectively learn the underlying dynamics of granular materials and provide accurate predictions for their motion. By integrating physical constraints into the neural network architecture, PINNs offer a powerful tool to model and simulate granular material behavior.

This thesis employs physics-informed neural networks in three distinct yet interconnected sections.

The first study focused on investigating the motion of particles within the bed of a biomass combustion chamber using physical-informed neural networks (PINNs). To obtain the necessary data to train the neural network, the discrete element method (DEM) simulations are employed with the XDEM software, and these simulation results were used as the

dataset. PINNs were chosen due to their ability to incorporate the governing equations into the loss function, allowing the neural network to handle the underlying physics of the problem, without having the information on the boundary conditions, initial conditions, and geometry. The neural network architecture was carefully fine-tuned to prevent overfitting and underfitting, resulting in a configuration of 10 hidden layers with 200 neurons per layer. The dataset consisted of approximately 2222 particles in 600 time steps, covering a simulation time of 60 seconds with a time step of 0.1.

Upon evaluation, the trained neural network exhibited excellent agreement with the exact data obtained from the DEM simulations. The average percent relative error for the predicted velocities was consistently below 1%, with only a slight deviation exceeding 0.1%. Additionally, the mean squared error demonstrated consistently low values, further confirming the reliability of the PINN in accurately predicting and reconstructing the velocity fields.

Moving on to the second study, the focus shifted to particle behavior in a moving grate combustion chamber interacting with the fluids. Two distinct piles of particles, Beach-wood and Stringy-wood, were observed in the combustion chamber, totaling 2222 particles.

In this study, the primary objective is to make assumptions about the motion of dense particles and, subsequently, reconstruct their approximate motion by using the Navier-Stokes and continuity equations. These equations provide a framework for understanding the behavior of fluids. By incorporating these equations into the analysis, it becomes possible to treat the motion of the particles like fluid motion, enabling the development of an estimated representation of their trajectories. The datasets were extracted from scattered particle data and mapped into the computational fluid dynamics (CFD) meshed region.

The PINN was employed to reconstruct the particles' fields, taking coordinates (x, y) and time (t) as inputs and predicting velocity (u, v), pressure (P), and density (ρ) fields. After tuning the neural network architecture, consisting of six layers with thirty neurons, the trained network was validated against numerical simulations. The predictions exhibited a high degree of agreement with the actual data, with velocity predictions in the x-direction

achieving approximately 1% error, velocity predictions in the y-direction approximately 2% error, pressure predictions approximately 0.6% error, and density predictions approximately 1.5% error.

Finally, the third study explored particle behavior and positions for the time more than training time. The trained neural network successfully predicted particle positions throughout the two case studies, the moving grate combustion camber, and mixing drum.

The results from two case studies, the moving grate, and the mixing drum, show the efficacy of neural network methods in predicting particle behavior and positions for time more than training time. In the moving grate study, the neural network accurately predicted particle positions beyond the training time, with low mean squared error (MSE) values for velocities and a relative percent error (RPE) below 5%. Similarly, in the mixing drum study, the neural network successfully predicted particle positions throughout the mixing process, achieving low MSE values below 0.6%. The comparison of loss values during training highlighted the network's learning process and improved prediction performance. The adoption of recurrent neural networks (RNNs) and physics-informed neural networks proved beneficial in capturing temporal dependencies and achieving accurate predictions. Visual assessments confirmed the physically plausible distribution of predicted particle positions, even when extended beyond the training period.

In summary, these combined studies showcased the effectiveness of physics-informed neural networks, in studying particle behavior and positions. The low error values obtained for particle velocities and positions validated the accuracy and reliability of the neural network models. This research contributes to the advancement of predictive modeling techniques for particle systems, benefiting various fields such as industrial processes, material science, and environmental engineering. Furthermore, the successful application of physics-informed deep learning in mechanical applications opens new avenues for exploration and innovation.

References

[1] Maw Maw Tun et al. “Biomass energy: An overview of biomass sources, energy potential, and management in Southeast Asian countries”. In: *Resources* 8.2 (2019), p. 81.

[2] Liang Xu et al. “Changes in global terrestrial live biomass over the 21st century”. In: *Science Advances* 7.27 (2021), eabe9829.

[3] *Biomass Resources*. https://www.alternative-energy-tutorials.com/biomass/biomass-resources.html.

[4] J Werther et al. “Combustion of agricultural residues”. In: *Progress in energy and combustion science* 26.1 (2000), pp. 1–27.

[5] *Drum Roller/Mixer*. https://www.materialshandling.com.au/products/drum-roller-mixer/.

[6] Jin Wang, Xiaoke Ku, and Shuna Yang. “Simulation of Biomass Pyrolysis in a Rotary Drum by Coupling CFD-DEM with a One-Dimensional Thermally Thick Model”. In: *Energy & Fuels* 36.7 (2022), pp. 3665–3679.

[7] Aivars Kaķītis and Imants Nulle. “Evaluation of drum feeders for biomass dosage”. In: *Engineering for Rural Development. Proceedings of the Inernational Scientific Conference (Latvia)*. LLU. 2008.

[8] George Em Karniadakis et al. “Physics-informed machine learning”. In: *Nature Reviews Physics* 3.6 (2021), pp. 422–440.

[9] Pushan Sharma et al. “A Review of Physics-Informed Machine Learning in Fluid Mechanics”. In: *Energies* 16.5 (2023), p. 2343.

[10] Shenghan Guo et al. “Machine learning for metal additive manufacturing: Towards a physics-informed data-driven paradigm”. In: *Journal of Manufacturing Systems* 62 (2022), pp. 145–163.

[11] Salvatore Cuomo et al. “Scientific machine learning through physics–informed neural networks: where we are and what’s next”. In: *Journal of Scientific Computing* 92.3 (2022), p. 88.

[12] Chen Cheng and Guang-Tao Zhang. “Deep learning method based on physics informed neural network with resnet block for solving fluid flow problems”. In: *Water* 13.4 (2021), p. 423.

[13] Fateme Darlik, Prasad Adhav, and Bernhard Peters. “Prediction of the biomass particles through the physics informed neural network.” In: *ECCOMAS Congress 2022-8th European Congress on Computational Methods in Applied Sciences and Engineering*. 2022.

[14] Fateme Darlik and Bernhard Peters. “Reconstruct the biomass particles fields in the particle-fluid problem using continuum methods by applying the physics-informed neural network”. In: *Results in Engineering* 17 (2023), p. 100917.

[15] Navid Aminnia et al. “Three-dimensional CFD-DEM simulation of raceway transport phenomena in a blast furnace”. In: *Fuel* 334 (2023), p. 126574.

[16] Ayhan Demirbas. “Potential applications of renewable energy sources, biomass combustion problems in boiler power systems and combustion related environmental issues”. In: *Progress in energy and combustion science* 31.2 (2005), pp. 171–192.

[17] Mikko Hupa, Oskar Karlström, and Emil Vainio. “Biomass combustion technology development–It is all about chemical details”. In: *Proceedings of the Combustion institute* 36.1 (2017), pp. 113–134.

[18] Magali RG Meireles, Paulo EM Almeida, and Marcelo Godoy Simes. “A comprehensive review for industrial applicability of artificial neural networks”. In: *IEEE transactions on industrial electronics* 50.3 (2003), pp. 585–601.

[19] Zhi-Hua Zhou and Ji Feng. “Deep Forest: Towards An Alternative to Deep Neural Networks.” In: *IJCAI*. 2017, pp. 3553–3559.

[20] Osama Moselhi and Tariq Shehab-Eldeen. “Classification of defects in sewer pipes using neural networks”. In: *Journal of infrastructure systems* 6.3 (2000), pp. 97–104.

[21] John J Hopfield. “Neural networks and physical systems with emergent collective computational abilities.” In: *Proceedings of the national academy of sciences* 79.8 (1982), pp. 2554–2558.

[22] George Bebis and Michael Georgiopoulos. “Feed-forward neural networks”. In: *IEEE Potentials* 13.4 (1994), pp. 27–31.

[23] Keiron O’Shea and Ryan Nash. “An introduction to convolutional neural networks”. In: *arXiv preprint arXiv:1511.08458* (2015).

[24] Wojciech Zaremba, Ilya Sutskever, and Oriol Vinyals. “Recurrent neural network regularization”. In: *arXiv preprint arXiv:1409.2329* (2014).

[25] Maziar Raissi. “Deep hidden physics models: Deep learning of nonlinear partial differential equations”. In: *The Journal of Machine Learning Research* 19.1 (2018), pp. 932–955.

[26] Chengping Rao, Hao Sun, and Yang Liu. “Physics-informed deep learning for incompressible laminar flows”. In: *Theoretical and Applied Mechanics Letters* 10.3 (2020), pp. 207–212.

[27] Maziar Raissi, Paris Perdikaris, and George E Karniadakis. “Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations”. In: *Journal of Computational physics* 378 (2019), pp. 686–707.

[28] Maziar Raissi, Paris Perdikaris, and George Em Karniadakis. “Physics informed deep learning (part i): Data-driven solutions of nonlinear partial differential equations”. In: *arXiv preprint arXiv:1711.10561* (2017).

[29] Shengze Cai et al. “Physics-informed neural networks (PINNs) for fluid mechanics: A review”. In: *Acta Mechanica Sinica* (2022), pp. 1–12.

[30] Maziar Raissi and George Em Karniadakis. “Hidden physics models: Machine learning of nonlinear partial differential equations”. In: *Journal of Computational Physics* 357 (2018), pp. 125–141.

[31] Li Wang and Zhenya Yan. “Data-driven rogue waves and parameter discovery in the defocusing nonlinear Schrödinger equation with a potential using the PINN deep learning”. In: *Physics Letters A* 404 (2021), p. 127408.

[32] Yo Nakamura et al. “Physics-Informed Neural Network with Variable Initial Conditions”. In: *Proceedings of the 7th World Congress on Mechanical, Chemical, and Material Engineering (MCM’21)* (2021). DOI: DOI:10.11159/htff21.113.

[33] Han Gao, Luning Sun, and Jian-Xun Wang. “PhyGeoNet: Physics-informed geometry adaptive convolutional neural networks for solving parameterized steady-state PDEs on irregular domain”. In: *Journal of Computational Physics* 428 (2021), p. 110079.

[34] Jan Oldenburg et al. “Geometry aware physics informed neural network surrogate for solving Navier-Stokes equation (GAPINN)”. In: (2022).

[35] Bernhard Peters et al. “XDEM multi-physics and multi-scale simulation technology: Review of DEM–CFD coupling, methodology and engineering applications”. In: *Particuology* 44 (2019), pp. 176–193.

[36] Amir Houshang Mahmoudi et al. “An experimental and numerical study of wood combustion in a fixed bed using Euler–Lagrange approach (XDEM)”. In: *Fuel* 150 (2015), pp. 573–582.

[37] Mohammad Mohseni and Bernhard Peters. “Effects of particle size distribution on drying characteristics in a drum by XDEM: A case study”. In: *Chemical Engineering Science* 152 (2016), pp. 689–698.

[38] Amir Houshang Mahmoudi, Florian Hoffmann, and Bernhard Peters. “Detailed numerical modeling of pyrolysis in a heterogeneous packed bed using XDEM”. In: *Journal of analytical and applied pyrolysis* 106 (2014), pp. 9–20.

[39] Amir Houshang Mahmoudi, Florian Hoffmann, and Bernhard Peters. “Semi-resolved modeling of heat-up, drying and pyrolysis of biomass solid particles as a new feature in XDEM”. In: *Applied Thermal Engineering* 93 (2016), pp. 1091–1104.

[40] Amir Houshang Mahmoudi et al. “Modeling of the biomass combustion on a forward acting grate using XDEM”. In: *Chemical engineering science* 142 (2016), pp. 32–41.

[41] 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.

[42] Diederik P Kingma and Jimmy Ba. “Adam: A method for stochastic optimization”. In: *arXiv preprint arXiv:1412.6980* (2014).

[43] Nico Bauer et al. “Global energy sector emission reductions and bioenergy use: overview of the bioenergy demand phase of the EMF-33 model comparison”. In: *Climatic Change* 163.3 (2020), pp. 1553–1568.

[44] Gabriel Barroso, Simon Roth, and Thomas Nussbaumer. “Investigation of biomass conversion on a moving grate by pyrolysis gas analysis and fuel bed modelling”. In: *Energy* 174 (2019), pp. 897–910.

[45] Husni Husin et al. “Combustion efficiency in a fluidized-bed combustor with a modified perforated plate for air distribution”. In: *Processes* 9.9 (2021), p. 1489.

[46] Leonel JR Nunes. “Biomass gasification as an industrial process with effective proof-of-concept: A comprehensive review on technologies, processes and future developments”. In: *Results in Engineering* (2022), p. 100408.

[47] Gloria Boafo-Mensah, Kwadwo Mensah Darkwa, and Gabriel Laryea. “Effect of combustion chamber material on the performance of an improved biomass cookstove”. In: *Case Studies in Thermal Engineering* 21 (2020), p. 100688.

[48] Zihong Xia et al. “Two-fluid simulation of moving grate waste incinerator: Comparison of 2D and 3D bed models”. In: *Energy* 216 (2021), p. 119257.

[49] Jingyuan Zhang. “Computational fluid dynamics (CFD) modeling for biomass and wate to energy production”. In: (2021).

[50] Michael C Baker et al. “Direct comparison of Eulerian–Eulerian and Eulerian–Lagrangian simulations for particle-laden vertical channel flow”. In: *AIChE Journal* 66.7 (2020), e16230.

[51] Salar Tavakkol et al. “An Eulerian-Lagrangian method for wet biomass carbonization in rotary kiln reactors”. In: *Renewable and Sustainable Energy Reviews* 139 (2021), p. 110582.

[52] Yaxiong Yu et al. “A model to improve granular temperature in CFD-DEM simulations”. In: *Energies* 13.18 (2020), p. 4730.

[53] JK Tanui et al. “Application of CFD-DEM method in modeling of wood combustion in a fixed bed”. In: *Bulgarian Chemical Communications* 52 (2020), pp. 396–403.

[54] LM Yao et al. “An optimized CFD-DEM method for fluid-particle coupling dynamics analysis”. In: *International Journal of Mechanical Sciences* 174 (2020), p. 105503.

[55] A Alexiadis et al. “The duality between particle methods and artificial neural networks”. In: *Scientific reports* 10.1 (2020), pp. 1–7.

[56] Shinya Inazumi et al. “Artificial intelligence system for supporting soil classification”. In: *Results in Engineering* 8 (2020), p. 100188.

[57] Hanbin Zhong et al. “CFD-based reduced-order modeling of fluidized-bed biomass fast pyrolysis using artificial neural network”. In: *Renewable Energy* 152 (2020), pp. 613–626.

[58] Rajat Neelam et al. “Mechanical response of additively manufactured foam: A machine learning approach”. In: *Results in Engineering* 16 (2022), p. 100801.

[59] Krittin Korkerd et al. “Artificial neural network model for predicting minimum fluidization velocity and maximum pressure drop of gas fluidized bed with different particle size distributions”. In: *South African Journal of Chemical Engineering* 37 (2021), pp. 61–73.

[60] Dipal Baruah, DC Baruah, and MK Hazarika. “Artificial neural network based modeling of biomass gasification in fixed bed downdraft gasifiers”. In: *Biomass and Bioenergy* 98 (2017), pp. 264–271.

[61] Ibham Veza et al. “Improved prediction accuracy of biomass heating value using proximate analysis with various ANN training algorithms”. In: *Results in Engineering* 16 (2022), p. 100688.

[62] Shashank Reddy Vadyala et al. “A review of physics-based machine learning in civil engineering”. In: *Results in Engineering* (2021), p. 100316.

[63] Maziar Raissi et al. “Deep learning of vortex-induced vibrations”. In: *Journal of Fluid Mechanics* 861 (2019), pp. 119–137.

[64] Syifaul Huzni et al. “Physics-based surrogate model for reinforced concrete corrosion simulation”. In: *Results in Engineering* 16 (2022), p. 100659.

[65] Erin McGowan, Vidita Gawade, and Weihong Guo. “A Physics-Informed Convolutional Neural Network with Custom Loss Functions for Porosity Prediction in Laser Metal Deposition”. In: *Sensors* 22.2 (2022), p. 494.

[66] Qiming Zhu, Zeliang Liu, and Jinhui Yan. “Machine learning for metal additive manufacturing: predicting temperature and melt pool fluid dynamics using physics-informed neural networks”. In: *Computational Mechanics* 67.2 (2021), pp. 619–635.

[67] Hui Pang et al. “Physics-informed neural network approach for heat generation rate estimation of lithium-ion battery under various driving conditions”. In: *Journal of Energy Chemistry* (2022).

[68] Stefano Buoso, Thomas Joyce, and Sebastian Kozerke. “Personalising left-ventricular biophysical models of the heart using parametric physics-informed neural networks”. In: *Medical Image Analysis* 71 (2021), p. 102066.

[69] E Onate and J Rojek. “Combination of discrete element and finite element methods for dynamic analysis of geomechanics problems”. In: *Computer methods in applied mechanics and engineering* 193.27-29 (2004), pp. 3087–3128.

[70] Catherine O’Sullivan and Jonathan D Bray. “Selecting a suitable time step for discrete element simulations that use the central difference time integration scheme”. In: *Engineering Computations* (2004).

[71] Bernhard Peters et al. “Process analysis in thermal process engineering with high performance computing using the example of grate firing”. In: *Proceedings of the 12th European Conference on Industrial Furnaces and Boilers*. 2020.

[72] Bernhard Peters. “The extended discrete element method (XDEM) for multi-physics applications”. In: *Scholarly Journal of Engineering Research* (2013).

[73] Bernhard Peters et al. “A discrete/continuous numerical approach to multi-physics”. In: *IFAC-PapersOnLine* 48.1 (2015), pp. 645–650.

[74] Fenglei Qi et al. “DEM simulation of dense granular flows in a vane shear cell: Kinematics and rheological laws”. In: *Powder Technology* 366 (2020), pp. 722–735.

[75] Van Hieu Bui, Minh Duc Bui, and Peter Rutschmann. “Combination of Discrete Element Method and Artificial Neural Network for Predicting Porosity of Gravel-Bed River”. In: *Water* 11.7 (2019), p. 1461.

[76] B Vaferi et al. “Artificial neural network approach for prediction of thermal behavior of nanofluids flowing through circular tubes”. In: *Powder technology* 267 (2014), pp. 1–10.

[77] Jeremy Yu et al. “Gradient-enhanced physics-informed neural networks for forward and inverse PDE problems”. In: *Computer Methods in Applied Mechanics and Engineering* 393 (2022), p. 114823.

[78] Alexander Henkes, Henning Wessels, and Rolf Mahnken. “Physics informed neural networks for continuum micromechanics”. In: *Computer Methods in Applied Mechanics and Engineering* 393 (2022), p. 114790.

[79] Shahed Rezaei et al. “A mixed formulation for physics-informed neural networks as a potential solver for engineering problems in heterogeneous domains: comparison with finite element method”. In: *Computer Methods in Applied Mechanics and Engineering* 401 (2022), p. 115616.

[80] Atilim Gunes Baydin et al. “Automatic differentiation in machine learning: a survey”. In: *Journal of Marchine Learning Research* 18 (2018), pp. 1–43.

[81] L Fletcher et al. “Optimizing the number of hidden nodes of a feedforward artificial neural network”. In: *1998 IEEE International Joint Conference on Neural Networks Proceedings. IEEE World Congress on Computational Intelligence (Cat. No. 98CH36227)*. Vol. 2. IEEE. 1998, pp. 1608–1612.

[82] Fateme Darlik and Bernhard Peters. “Using the trained recurrent neural network to predict particles position for time more than training time”. In: (Sept. 2023). Unpublished Manuscript.

[83] HP Zhu et al. “Discrete particle simulation of particulate systems: a review of major applications and findings”. In: *Chemical Engineering Science* 63.23 (2008), pp. 5728–5770.

[84] Ganggang Xu et al. “Effect of drum structure on particle mixing behavior based on DEM method”. In: *Particuology* 74 (2023), pp. 74–91.

[85] Mengxiao Yu et al. “Three-dimensional DEM simulation of polydisperse particle flow in rolling mode rotating drum”. In: *Powder Technology* 396 (2022), pp. 626–636.

[86] JQ Gan, ZY Zhou, and AB Yu. “A GPU-based DEM approach for modelling of particulate systems”. In: *Powder Technology* 301 (2016), pp. 1172–1182.

[87] Fani Boukouvala et al. “Reduced-order discrete element method modeling”. In: *Chemical Engineering Science* 95 (2013), pp. 12–26.

[88] Yong Yu et al. “A review of recurrent neural networks: LSTM cells and network architectures”. In: *Neural computation* 31.7 (2019), pp. 1235–1270.

[89] Prayagraj Singh Deora et al. “Biofuels: An alternative to conventional fuel and energy source”. In: *Materials Today: Proceedings* 48 (2022), pp. 1178–1184.

[90] Mohammad Hosseini Rahdar, Fuzhan Nasiri, and Bruno Lee. “A review of numerical modeling and experimental analysis of combustion in moving grate biomass combustors”. In: *Energy & Fuels* 33.10 (2019), pp. 9367–9402.

[91] Ignazio Blanco. “The use of composite materials in 3D printing”. In: *Journal of Composites Science* 4.2 (2020), p. 42.

[92] Shunying Ji, Siqiang Wang, and Zongyan Zhou. “Influence of particle shape on mixing rate in rotating drums based on super-quadric DEM simulations”. In: *Advanced Powder Technology* 31.8 (2020), pp. 3540–3550.

[93] Jakub Hlosta et al. “DEM investigation of the influence of particulate properties and operating conditions on the mixing process in rotary drums: Part 1—Determination of the DEM parameters and calibration process”. In: *Processes* 8.2 (2020), p. 222.

[94] Xavier Besseron, Henrik Rusche, and Bernhard Peters. “Parallel Multi-Physics Simulation of Biomass Furnace and Cloud-based Workflow for SMEs”. In: *Practice and Experience in Advanced Research Computing*. 2022, pp. 1–9.

[95] Peter A Cundall and Otto DL Strack. “A discrete numerical model for granular assemblies”. In: *geotechnique* 29.1 (1979), pp. 47–65.

[96] V Aleshin and Koen Van Den Abeele. “Preisach analysis of the Hertz–Mindlin system”. In: *Journal of the Mechanics and Physics of Solids* 57.4 (2009), pp. 657–672.

[97] Yutaka Tsuji, Toshitsugu Tanaka, and T Ishida. “Lagrangian numerical simulation of plug flow of cohesionless particles in a horizontal pipe”. In: *Powder technology* 71.3 (1992), pp. 239–250.

[98] Felipe AC Viana et al. “Estimating model inadequacy in ordinary differential equations with physics-informed neural networks”. In: *Computers & Structures* 245 (2021), p. 106458.

[99] Guang Chen. “Recurrent neural networks (RNNs) learn the constitutive law of viscoelasticity”. In: *Computational Mechanics* 67.3 (2021), pp. 1009–1019.

[100] Ashraf Darwish, Aboul Ella Hassanien, and Swagatam Das. “A survey of swarm and evolutionary computing approaches for deep learning”. In: *Artificial intelligence review* 53 (2020), pp. 1767–1812.

[101] Tomasz Szandała. “Review and comparison of commonly used activation functions for deep neural networks”. In: *Bio-inspired neurocomputing* (2021), pp. 203–224.

[102] Jacob Hilton, Karl Cobbe, and John Schulman. “Batch size-invariance for policy optimization”. In: *Advances in Neural Information Processing Systems* 35 (2022), pp. 17086–17098.

[103] Bagus Tris Atmaja and Masato Akagi. “Evaluation of error-and correlation-based loss functions for multitask learning dimensional speech emotion recognition”. In: *Journal of Physics: Conference Series*. Vol. 1896. 1. IOP Publishing. 2021, p. 012004.

[104] Qi Zhang et al. “Comprehensive assessment of energy conservation and CO2 emissions mitigation in China’s iron and steel industry based on dynamic material flows”. In: *Applied Energy* 209 (2018), pp. 251–265.

[105] Jose Adilson de Castro, Cyro Takano, and Jun-ichiro Yagi. “A theoretical study using the multiphase numerical simulation technique for effective use of H2 as blast furnaces fuel”. In: *Journal of Materials Research and Technology* 6.3 (2017), pp. 258–270.

[106] XU Chunbao Charles and Da-qiang Cang. “A brief overview of low CO2 emission technologies for iron and steel making”. In: *Journal of Iron and Steel Research, International* 17.3 (2010), pp. 1–7.

[107] Wenqiang Sun et al. “Material and energy flows of the iron and steel industry: Status quo, challenges and perspectives”. In: *Applied Energy* 268 (2020), p. 114946.

[108] Yansong Shen et al. “Three-dimensional modeling of flow and thermochemical behavior in a blast furnace”. In: *Metallurgical and Materials Transactions B* 46.1 (2015), pp. 432–448.

[109] S Dong et al. “Characterization of tuyere-level core-drill coke samples from blast furnace operation”. In: *Energy & fuels* 21.6 (2007), pp. 3446–3454.

[110] Yoshiyuki MATSUI et al. “Analyses on blast furnace raceway formation by micro wave reflection gunned through tuyere”. In: *ISIJ international* 45.10 (2005), pp. 1432–1438.

[111] GSSRK Sastry, GS Gupta, and AK Lahiri. “Cold model study of raceway under mixed particle conditions”. In: *Ironmaking & steelmaking* 30.1 (2003), pp. 61–65.

[112] Sastry GSSRK. “Void formation and breaking in a packed bed”. In: *ISIJ international* 43.2 (2003), pp. 153–160.

[113] Ruixuan Zhang et al. “Image-based flame detection and combustion analysis for blast furnace raceway”. In: *IEEE Transactions on instrumentation and measurement* 68.4 (2019), pp. 1120–1131.

[114] SS Mondal, SK Som, and SK Dash. “Numerical predictions on the influences of the air blast velocity, initial bed porosity and bed height on the shape and size of raceway zone in a blast furnace”. In: *Journal of Physics D: Applied Physics* 38.8 (2005), p. 1301.

[115] Xiaoping Qiu et al. “A simplified two-fluid model coupled with EMMS drag for gas-solid flows”. In: *Powder technology* 314 (2017), pp. 299–314.

[116] WJ Yang, ZY Zhou, and AB Yu. “Discrete particle simulation of solid flow in a three-dimensional blast furnace sector model”. In: *Chemical Engineering Journal* 278 (2015), pp. 339–352.

[117] BH Xu et al. “A numerical and experimental study of the gas-solid flow in a fluid bed reactor”. In: *Powder Handling & Processing* 13.1 (2001), pp. 71–76.

[118] Yu-Qing Feng et al. “Numerical study of gas-solid flow in the raceway of a blast furnace”. In: *steel research international* 74.9 (2003), pp. 523–530.

[119] Shinichi Yuu, Toshihiko Umekage, and Masatomo Kadowaki. “Numerical simulation of particle and air velocity fields in raceway in model blast furnace and comparison with experimental data (cold model)”. In: *ISIJ international* 50.8 (2010), pp. 1107–1116.

[120] JE Hilton and PW Cleary. “Raceway formation in laterally gas-driven particle beds”. In: *Chemical engineering science* 80 (2012), pp. 306–316.

[121] Shuai Wang and Yansong Shen. “CFD-DEM modelling of raceway dynamics and coke combustion in an ironmaking blast furnace”. In: *Fuel* 302 (2021), p. 121167.

[122] Zhen Miao et al. “CFD-DEM simulation of raceway formation in an ironmaking blast furnace”. In: *Powder technology* 314 (2017), pp. 542–549.

[123] Jiaxin Cui, Qinfu Hou, and Yansong Shen. “CFD-DEM study of coke combustion in the raceway cavity of an ironmaking blast furnace”. In: *Powder Technology* 362 (2020), pp. 539–549.

[124] E Dianyu et al. “Particle-scale study of coke combustion in the raceway of an ironmaking blast furnace”. In: *Fuel* 311 (2022), p. 122490.

[125] Yuting Zhuo, Zhongjie Hu, and Yansong Shen. “CFD study of hydrogen injection through tuyeres into ironmaking blast furnaces”. In: *Fuel* 302 (2021), p. 120804.

[126] Tao Li et al. “Numerical Simulation Study on the Effects of Co-Injection of Pulverized Coal and Hydrochar into the Blast Furnace”. In: *Sustainability* 14.8 (2022), p. 4407.

[127] Tyamo Okosun, Samuel Nielson, and Chenn Zhou. “Blast Furnace Hydrogen Injection: Investigating Impacts and Feasibility with Computational Fluid Dynamics”. In: *JOM* 74.4 (2022), pp. 1521–1532.

[128] Mehdi Baniasadi, Maryam Baniasadi, and Bernhard Peters. “Coupled CFD-DEM with heat and mass transfer to investigate the melting of a granular packed bed”. In: *Chemical Engineering Science* 178 (2018), pp. 136–145.

[129] Bernhard Peters. *Thermal conversion of solid fuels*. eng. International series on developments in heat transfer Vol. 15 15. Southampton: WIT Press, 2003. ISBN: 1-85312-953-4.

[130] Thomas Fred Eric Rhead and Richard Vernon Wheeler. “CCXXVI.—The effect of temperature on the equilibrium 2CO⇌CO 2+ C”. In: *Journal of the Chemical Society, Transactions* 97 (1910), pp. 2178–2189.

[131] Tao Zhao. *Coupled DEM-CFD analyses of landslide-induced debris flows*. Springer, 2017.

[132] Dimitri Gidaspow. *Multiphase flow and fluidization: continuum and kinetic theory descriptions*. Academic press, 1994.

[133] CY Wen and YH Yu. “Mechanics of fluidization”. In: *The Chemical Engineering Progress Symposium Series* 162 (1966), pp. 100–111.

[134] Sabri Ergun and Ao Ao Orning. “Fluid flow through randomly packed columns and fluidized beds”. In: *Industrial & Engineering Chemistry* 41.6 (1949), pp. 1179–1184.

[135] Heng Xiao and Jin Sun. “Algorithms in a robust hybrid CFD-DEM solver for particle-laden flows”. In: *Communications in Computational Physics* 9.2 (2011), pp. 297–323.

[136] Edder Rabadan Santana, Gabriele Pozzetti, and Bernhard Peters. “Application of a dual-grid multiscale CFD-DEM coupling method to model the raceway dynamics in packed bed reactors”. In: *Chemical Engineering Science* 205 (2019), pp. 46–57.

[137] Hiroshi Nogami, Hideyuki Yamaoka, and Kouji Takatani. “Raceway design for the innovative blast furnace”. In: *ISIJ international* 44.12 (2004), pp. 2150–2158.