Agata Widuch

Development novel approaches for modeling dense granular flows

Doctoral dissertation

Supervisor:
Prof. Ph. D. DSc Wojciech Adamczyk

Science discipline:
Environmental Engineering, Mining and Energy

This research is supported by National Science Centre Poland within projects UMO-2018/31/B/ST8/02201. This help is gratefully acknowledged herewith.

Silesian University of Technology

Gliwice, 2023

Author
Agata Widuch
Silesian University of Technology
Faculty of Energy and Environmeental Engineering
Department of Thermal Technology
Konarskiego St. 22
44-100 Gliwice
Poland
e-mail: agata.widuch@polsl.pl

Contents

Acknowledgments

I am profoundly grateful to my supervisor, Prof. Ph. D. DSc. Wojciech Adamczyk, for his immense patience, belief in my abilities, open and enriching discussions we had during our meetings. I am also thankful for support in exploring new ideas and sharing his wealth of knowledge. I would also like to thank Ph. D. Pawel Morkisz, from AGH University of Krakow, for his patience, openness and the opportunity to grow in a new environment.

I would like to thank the research team from Department of Thermal Technology, and Lappeenranta University of Technology, especially Ph. D. Marcin Nowak for their cooperation.

This scientific work in the presented form would not have been possible without the financial assistance of the National Science Center, within project (A novel approach for modeling of complex granular flows) No 2018/31/B/ST8/02201 and partially by the National Centre for Research and Development, within project (Development and demonstration of a computer system for controlling operation and managing the availability and reliability of industrial infrastructure based on artificial intelligence algorithms - OPTI_AI_UNIT) POIR.01.01.01-00-1253/19-00. The support is gratefully acknowledged herewith.

Finally, I am deeply grateful to my Family, friends for accompanying me during my studies. Without your support, completing this journey would have been impossible.

The work is dedicated to beloved Parents.

Agata Widuch

1
Introduction

1.1. Fluidization

Fluidization is a multiphase process, where solid material is transported by a flowing gas. Over the past few decades, the fluidization process has garnered increasing attention. It finds diverse uses in various industries, including chemical applications for re-drying chemicals post-mechanical drying [43], pharmaceutical applications to enhance medication mixing efficiency and achieve uniform material distribution [39], and in the food processing industry, such as the operation of a popcorn popper [126]. Additionally, the technology has numerous applications in the energy sector due to many advantages over traditional boiler constructions. All this is mostly of increasing calculation resources which allows the application of advance computer models to predict behavior of solid phase.

The last mentioned application is undergoing significant development. The key attributes of this technology include intense heat transfer, effective mixing of solid materials, extended fuel residence time, and uniform temperature distribution throughout the combustion chamber that allows easier control of NOx emission and eliminates thermal NOx generation due to elimination of local high temperature picks. Industrial boilers utilizing fluidization techniques exhibit similar boiler efficiency, compared to pulverized coal boilers but have an advantage in case of decreased harmful gases emissions [29]. This is mainly attributed to the rapid mixing rate, which positively impacts all thermochemical processes involved in coal combustion. One example is the amount of SOx emitted, which is significantly reduced when using fluidized bed boilers, through direct limestone injection [105].

In the fluidized bed, the combustion temperature typically falls within the range of $800-950°C$ [82], resulting in reduced NOx emissions and enabling the combustion of fuels with lower heating values, including waste materials [125]. The lower temperature prevents the ash from softening, preventing issues related to slag or coating formation on chamber walls. At the same time it is possible to burn fuels with higher ash content [33]. As a result, the heat transfer rate between the heating surfaces and the heated medium remains unaffected. Another advantage of using boilers with fluidized bed, is their minimal sensitivity to fuel quality. A wide variety of fuel types, including waste materials, can be easily burnt. In terms of the material proportions within the combustion chamber, only 1-3% [82, 18] consists of coal fraction, while the majority comprises inert material, typically

sand [32]. The inert material serves the purpose of transferring the heat released during the combustion process [80].

Fluidization, as a basic process wherein the flow of gas through the chamber causes solid particles to act like a fluid-like state. In simple laboratory construction the process starts by placing the material on a porous plate. Subsequently, gas with a low flow rate passes through the porous plate and the material. The minimal fluidization velocity of the gas (u_{mf}) is attained when the forces acting on the particles, including gravity, are balanced, and they begin to move, as shown in Fig. 1.1A. With further increase in velocity above the minimal fluidization velocity, smaller particles rise to the top of the boiler, leading to the formation of gas bubbles, which can be seen in Fig. 1.1B. When the velocity exceeds, so-called terminal velocity (u_{tf}), the bubbles vanish, and instead, turbulent motion of particles occurs, what is depicted in Fig. 1.1C. As the gas velocity continues to rise, the turbulent bed transforms into a circulating fluidized bed. Particles are removed through the top part of the chamber, and after passing over the solid separator, they return to the chamber, as illustrated in Fig. 1.1D.

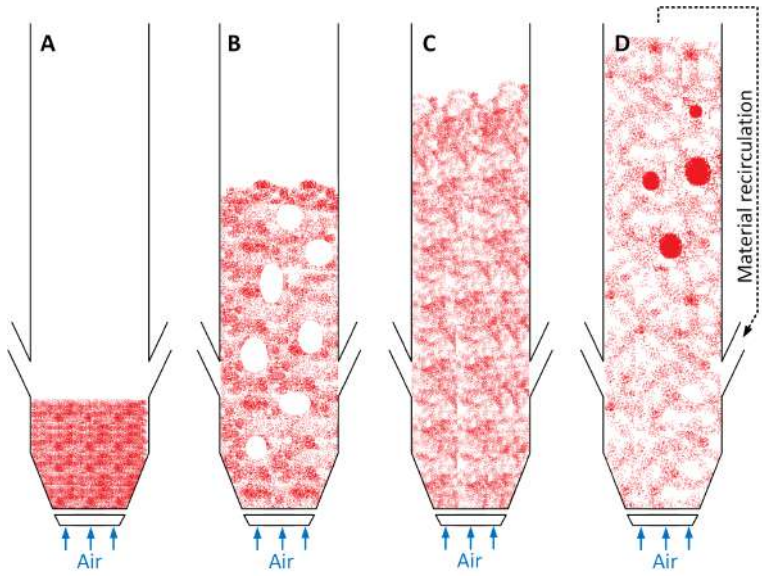

Fig. 1.1. Fluidization regimes, where **A**: Fixed bed, **B**: Bubbling bed, **C**: Turbulent bed, **D**: Circulating Fluidized bed

1.1.1. Fluidization regimes

In a fluidized bed boiler, three characteristics zones can be distinguished: the dense zone, transition zone, and dilute zone (refer to Figure 1.2). The greater the distance from the gas distributor the lower the concentration of particles. The dense zone marks the initiation of fluidized bed formation. Due to high volume concentration, particles are accumulating into so called clusters, where the interaction between solid material is an effect of friction forces. The distance between particles is so small, therefore the particle collisions are not significant in that zone. The clusters expand to the transition zone as a result of friction and kinetic transport. When the distance between particles in the higher part of a boiler increase, the collisions take over dominance. Kinetic transport starts to dominate, when the distance between particles gets large. Particle size and density, as well as overall mass of material in a boiler influences not only the minimum fluidization velocity but also the size of particular zones. Particles with smaller diameter are easily moving to the upper parts of a boiler.

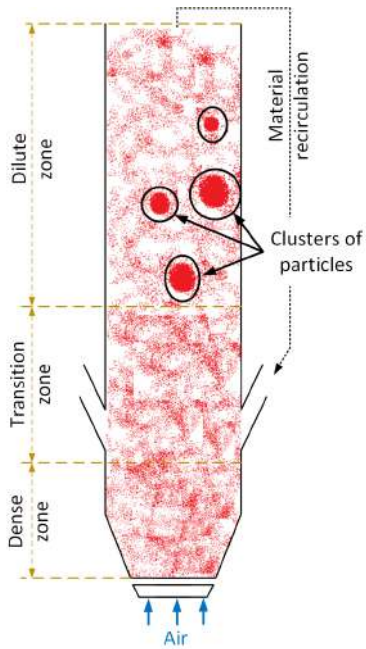

Fig. 1.2. Fluidization zones

1.1.2. Bubbling Fluidized Bed

Compared to Pulverized Coal Fired (PC) boilers, Bubbling Fluidized Bed (BFB) boilers operate at lower temperatures ranging from 800°C to 950°C [114]. The combustion chamber is the main part of the BFB boiler as it facilitates the conversion of fuel into heat. Depending on the desired output, the boiler can produce hot water or steam. The bed material consists of a mixture of fuel and inert material, a characteristic shared by all fluidized beds. Additionally, limestone is commonly added to aid in sulphur capture. [64].

Prior to initiating combustion, the bed material must be mixed with the fuel. The bed material is a mixture of fuel and inert material, which typically is a sand with the addition of limestone. Within the BFB boiler, there are two points where air is supplied: at the bottom of the chamber and in the upper part of combustion zone, marked on Figure 1.3. Adequate mixing of the bed material in the bottom of the boiler, is ensured by supplying primary air from the bottom of the chamber. The gas velocity constantly varies and is maintained within a range between the minimum fluidization velocity and the entrainment velocity [47]. Residues of burnt fuel in the form of small-diameter ash particles are removed from the combustion chamber through orifices. The most significant heat fluxes in BFB boilers are concentrated at the bottom. To protect bottom part from erosion and water dry-out, the internal surface is coated with appropriate refractory material [138].

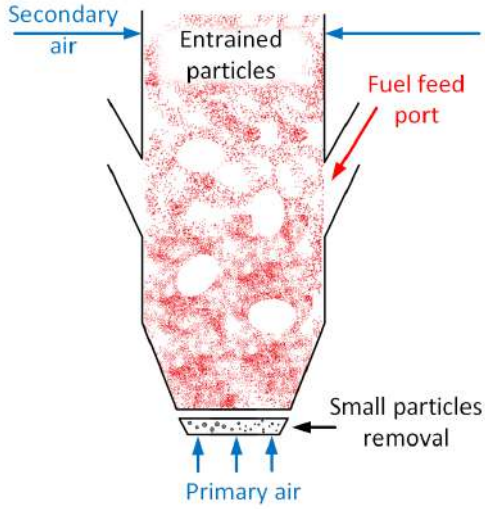

Fig. 1.3. Closure of the bottom of BFB boiler

1.1.3. Circulating Fludized Bed

Currently, the most popular fluidization technology is the Circulating Fluidized Bed (CFB). One of the main advantages of such boilers is their ability to handle various fuel qualities. The CFB boiler can efficiently burn fuels with higher moisture content and lower heating values, such as biomass or RDF (Refuse Derived Fuel) [115]. It becomes particularly significant in addressing the increasing challenges of waste management. CFB boilers are primarily utilized in high-power generating units. Several installed units, for example, have electrical power capacities of 460 MWe (Łagisza, Poland), 330 MWe (Novocherkasskaya, Russia), and 550 MWe (Samcheok, South Korea) [101].

The particle residence time in a closed loop is significantly longer in comparison to traditional pulverized coal boilers. Instead of removing unburnt fuel particles, they are separated from flue gases and returned to the combustion chamber through a separator and loop seal. The high conversion rate of fuel contributes to higher boiler efficiency. Furthermore, co-combustion of different types of fuels is feasible and can function effectively. Adjusting the amount of oxidizer provided to the combustion chamber in different zones is crucial and depends on the type of fuel, considering its size, density, and moisture content. Among the operating conditions, maintaining the velocity above the entrainment velocity, typically ranging from 4.5 to 6.7 m/s, is essential [127]. It uplifts particles and ensures their uniform redistribution within the chamber. Particles with smaller mass are raised, while heavier ones occupy the bottom part of the chamber until they fragment into smaller particles during collision, particle swell, and breakage processes. A typical CFB boiler configuration (visualization is presented on a Figure 1.4) includes the combustion chamber, loop-seal, and drain section.

The primary air is supplied by the oxidizer distributor (separator). The combustion chamber material is composed of fuel, solid material and sorbents. Injecting air at a speed surpassing the terminal velocity results in a highly turbulent bed. Another factor influencing turbulence is the shape of the boiler, often designed with a narrower lower part and a wider upper part. This arrangement continuously carries particles from the bottom to the top, and if they are still big and heavy enough are eventually returned to the boiler through the loop seal and drain section. The height to which particles rise depends on their mass. If gravitational force dominates, particles glide along the walls and fall back down. The combustion temperature is uniform and lower (800 °C-950 °C [97]), along with reduced heat fluxes to the boiler walls, thanks to the high heat transfer and mixing rates, in comparison to PC boilers. Similar to BFB boilers, the internal surface of CFB boiler walls is coated with refractory material to be shield from the high heat fluxes, occurring inside the combustion chamber.

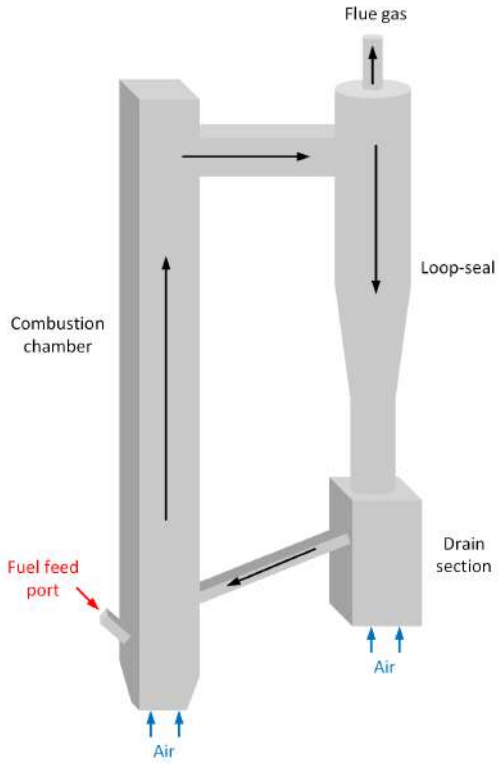

Fig. 1.4. Simplified general scheme of CFB boiler with highlighted most important parts

1.1.4. The particle hydrodynamics

The complexity of flows with two phases present (solid and gas), is attributed to the hydrodynamic interactions between those two. These interactions involve various forces, which can be categorized into different groups. The gravitational force is counteracted by the fluid drag and buoyancy forces, acting in opposite directions. When these forces reach a pseudo-equilibrium state, the particle becomes suspended in the carrying gas. At this point, the medium in the fluidized bed behaves as if it were a fluid, and it can be described using mathematical tools developed for dealing with continuous phases:

- the static pressure at any height in the vessel is approximately equal to the weight of the solid bed per unit of cross-sectional area, above the level, where the pressure acts,

- an object with higher density than the density of the bed sinks, while the lighter objects float following Archimedes principle [119],
- the solids from the bed can be discharged, similar to a liquid, through an orifice located either at the bottom or on the side of the container,
- the bed surface remains horizontal level, and the bed takes on the shape of the vessel,
- for properly mixed material, the bed exhibits a nearly uniform temperature when subjected to heating.

An illustrative example demonstrating the disparities between solid and fluid flow can be observed in an *hour-glass* and a *U-tube pipe*. In the hour-glass, the particle flow rate through the orifice at the bottom remains constant and is not influenced by the height of the particle bed. It is in contrast to fluids, where the flow rate through the orifice depends on the hydrostatic pressure. In the case of a U-tube filled with fluid, both arms contain the same amount of water. However, if sand is used instead of fluid, only one arm of the U-tube gets filled.

The pressure drop across the bed is caused by the drag forces acting on the particles in the moving fluid. The pressure drop per unit height of a packed bed ($\Delta p/H$) containing uniformly sized particles can be described using the Ergun equation [54]

$$\frac{\Delta p}{H} = \underbrace{\left[150\frac{(1-\varepsilon_f)^2}{\varepsilon_f^3}\frac{\mu_f}{(k_v d_p)^2} + 1.75\frac{\rho_f(1-\varepsilon_f)u_f}{\varepsilon_f^3 d_p k_v}\right]}_{K_{fs}} u_f \tag{1.1}$$

where term K_{fs} stands for the interphase exchange coefficient between phases, ε_f stands for the void fraction of the bed, d_p is the diameter of the particles, u_f determines the superficial velocity of the gaseous phase, i.e., the velocity of the fluid through an empty tube at the same volumetric flow rate, H defines the bed height and k_v is the particle sphericity which for ideal sphere is equal to $\pi/6$.

As the pressure drop in the solid bed increases, the superficial gas velocity u_f approaches a critical value known as the minimum fluidization velocity u_{mf}. The pressure drop for a bed at rest can be defined as follows [58]

$$\Delta p = H(1-\varepsilon_f)(\rho_p - \rho_f)g \tag{1.2}$$

where H stands for the height of the bed, ρ_p is the particle density and g determines the acceleration of gravity. The minimal value of fluid velocity u_{mf}, when the bed starts to fluidize, can be determined by solving simultaneously equations (1.1) and (1.2) assuming $u_f = u_{\mathrm{mf}}$.

Two other significant velocities, which are important in particle transport phenomena in a fluidized bed, are the terminal velocity u_{tf} and relative velocity u_{r}. The terminal velocity can be determined using the particle force balance equation (1.3), with the assumption that the gas velocity u_f is zero, and the particle velocity u_p is equal to the terminal velocity u_{tf}. The particle balance equation (1.3) was derived for the movement of a single particle under gravitational, buoyancy, and drag forces.

$$\underbrace{m_p g}_{\text{Gravity}} = \underbrace{m_p \frac{\rho_f g}{\rho_p}}_{\text{Buoyancy}} + \underbrace{C_{\mathrm{D}} \frac{\pi \left(u_f - u_p\right)^2 \rho_f}{8} d_p^2}_{\text{Drag}} \tag{1.3}$$

In equation (1.3), the term $(u_f - u_p)$ represents the relative velocity (or slip velocity, u_{r}), which accounts for the resistance of particles against falling, when the gas and particles move upwards. The parameter C_{D} stands for the drag coefficient.

In multiphase flows, where the solid phase is relatively diluted $(1 - \varepsilon_f) < 0.1$, one- and two-way coupling interactions between phases are involved [47]. If the interactions between particles can be neglected, one-way coupling occurs, where the gaseous phase affects the particle motion, but there is no reverse effect of the particle movement on the fluid. In cases where both directions of influence are considered, the coupling is referred to as two-way coupling.

1.2. Motivation and background

Whenever there is a need to construct new units or enhance existing ones, a significant amount of engineering effort becomes necessary. Ongoing environmental regulations consistently force the energy industry to innovate fresh methods, for transforming fossil fuels into forms of high energy. The atmospheric CO_2 concentration is continually rising primarily as a result of fossil fuel combustion, resulting in a rise in the overall temperature [51]. Therefore, it is necessary to look for alternative options. Fluidized beds could serve as a complement to renewable energy sources like wind turbines, solar panels, and hydropower stations, as these sources heavily rely on weather conditions. Despite being environmentally friendly solutions, their effectiveness remains relatively modest, ranging from 18% to 30.5% in efficiency [107, 79].

To avoid the need for conducting tests directly on site or in a laboratory environment, numerical methods can be employed. The utilization of Computational Fluid Dynamics (CFD) appears to be the most fitting choice. Fluidization is an example of a physical process causing engineers considerable challenges.

Primarily, such complexity arises from the mathematical models developed to describe phase interaction processes. To attain a high level of precision in the model, and to

establish suitable coefficients, experimental data is essential. Prior to implementing the model in large-scale boilers, preliminary experiments are necessary to validate the submodels. Moreover, the extensive application of the fluidization process, owing to its complexity, is regarded as one of the most demanding processes for mathematical modeling.

Developing a reliable mathematical model for simulating the fluidization process is not trivial, due to the involvement of chemical processes, varying spatial and temporal scales, intricate connections between gaseous and solid phases, and interplay among particles. The complexity is additionally, increased by the inclusion of combustion, gasification, particle fragmentation, attrition, and agglomeration. A significant hurdle is the requirement for a computing tool that's precise, swift, reliable, and consistent in predicting and modeling particle fluidization. The existing computational methods capable of addressing such complicated process are categorized into four groups:

- microscale - discrete element model, discrete numerical simulation, etc.
- macroscale - empirical and semi-empirical models,
- meso- and macroscale - multifluid models like Euler–Euler models,
- macroscale - hybrid Euler–Lagrange (HEL) models (known also as the multiphase particle-in-cell (MP-PIC) method)

As each of these methods has its own limitations, there is still a potential for improvement, in existing approaches, through advancements in numerical modeling capabilities. The field of engineering is poised to confront a significant challenge in the upcoming years: the development of a dependable, foresighted, resilient, and multi-scale modeling tool, able to enhance modeling capabilities, while minimizing computational demands. Computational techniques present a significant opportunity for foreseeing and comprehending the complex physical mechanisms, that underlie the fluidization process.

The accurate prediction of particle collisions enables the determination of optimal material distribution within a bed, and precise evaluation of pressure profiles, drag forces, and areas prone to erosion. A significant challenge in modeling the fluidization process revolves around the requirement for a precise, resilient, rapid, and stable numerical model to predict particle collisions within dense systems. Such systems are encountered in applications like fluidized bed boilers (used in waste management, energy, and environmental sectors) and pressurized solid gasifiers. Without such a model, achieving reliable predictions of solid distribution within the system becomes hard to reach. Consequently, it directly impacts projected pressure profiles, temperature distributions, reductions in harmful species, projected solid separation efficiency, erosion predictions, and more, as detailed in previous studies [42, 55, 13].

The already available approaches such as, s Euler-Euler (EE), hybrid Euler-Lagrange (HEL), and Discrete Element Model (DEM: Soft or Hard Sphere), encounter some main challenges. The most noteworthy among these are:

- lengthy computation time, when using the DEM approach,
- difficulty in predicting Granular Temperature (GT - Θ), a factor used in all closure terms for collisional and phase interaction models, in both EE and HEL methods,
- difficulties in predicting particle size distribution, when using the EE approach, as each particle size is represented by an additional solid phase,
- high instability in the solution, when the number of particles in the system is large (over 3 million), with the HEL approach,
- issues with mass conservation during interpolation from discrete phase to Eulerian grid when using HEL approach (partially addressed in the MP-PIC approach with commercial CPFD Software, LLC Barracuda code), and more.

To tackle above mentioned challenges, the engineering community, focused on particle transport modeling will require significant innovations in the coming years. These innovations are necessary to establish modeling approaches, that can predict fluidization with high accuracy, while maintaining reasonable computational efficiency, for resolving particle-particle interactions. Developing such an accurate model necessitates the adoption of a multiscale modeling strategy, for the fluidization process. It involves predicting particle collisions at the micro-scale level, while modeling particle transport at the meso- and macro-scales [95], as illustrated in Figure 1.5. As a solution, there is a room for a dependable and adaptable Surrogate Model (SM) capable of predicting particle-particle collisions and behavior. Such model would allow the application of numerical tools within a realistic time frame, for large fluidized systems.

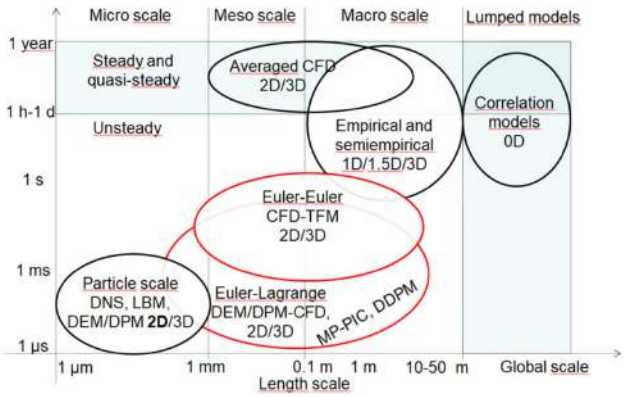

Fig. 1.5. Classification of multiphase models for fluidized systems [95]

Usually two main approaches are used for simulating fluidization processes. The first approach, is the Euler-Euler method, as described in [57, 58, 59]. In this technique, both continuous and dispersed phases are treated as interpenetrating continua. This approach was developed under the assumption, that the solid phase can be treated as a continuous phase with representative properties as the fluid [57]. The initial derivation of equations for two-phase systems can be found in the work of Anderson and Jackson [22]. When applying a two-fluid model to simulate particle transport phenomena, it is necessary to define closure models. These models serve the purpose of predicting particle collisions and fluid-solid interactions. The prediction of particle interactions can be accomplished using the principles of dense gases [38] and the kinetic theory of granular flow (KTGF) [124]. However, an additional drawback of the Eulerian model is the requirement for a fine mesh, to accurately model cluster formation. It results in extended computational time and influences the calculated pressure profile, along the combustion chamber.

An approach, commonly used to model particle interactions, involves the application of discrete particle techniques. These discrete models, when employed in dense dispersed phases within fluidized bed boilers, offer an accurate way of predicting interactions between particles, as well as particle-fluid interactions and Particle Size Distribution (PSD), within the dispersed phase. The Discrete Element Method (DEM) [47] is particularly suited for achieving such goal. Both of these approaches, the hard-sphere and soft-sphere collision models, are utilized to describe particle interactions. In the hard-sphere model, particle interactions are assumed to be instantaneous [67]. The effect of particle collisions is identified between each pair of particles, which can be computationally intensive. The soft-sphere model employs Hertzian contact theory, which models slight overlap between particles during contact [129]. However, due to the substantial computational need, the practicality of using DEM is limited to smaller scale problems. Certain applications of the hard-sphere DEM for modeling particle transport in small-scale fluidized beds, can be found in [143]. In the work of Tsuji et al. [128], the soft-sphere collision model was used, to simulate a bubbling fluidized bed, with 4.5 million particles. Despite the increasing computational power, DEM remains too numerically resource-expensive.

An alternative technique for modeling the fluidization process, which incorporates realistic Particle Size Distribution (PSD), is the hybrid Euler-Lagrange approach. In contrary to the Eulerian approach, it treats the fluid phase as a continuum, while the solid phase is tracked in a Lagrangian reference frame. To speed up numerical simulations while maintaining accuracy, a hybrid Euler-Lagrange approach was formulated by Andrews and O'Rourke [23]. This approach was subsequently refined by Patankar [108, 109] and Snider [122, 120, 121]. To simplify calculations, particles are packed into parcels with the same physical properties as mass, diameter, and density, reflecting particle behavior.

Particle-particle interactions are not directly simulated; instead, they are addressed through the kinetic theory of granular flow (KTGF) within the Eulerian frame, where the particles are projected [76].

Each of the above approaches has its drawbacks. Whether in the form of the time required to obtain results, hardware requirements, stability of calculations, magnitudes regarding the cases to be analyzed, etc., they all have their drawbacks. Therefore, there is still room and opportunity to try to develop another approach to simulate multiphase flows using new methods or other means. In this work, it was decided to use simplified models to replace the expressions responsible for the collision members in the equations, in the approaches already available. As a result, there is no significant decrease in the accuracy of the simulation, for that the calculation time will be reduced.

1.3. Objectives

Recognizing both the benefits and drawbacks of the EE, HEL, and DEM models for modeling granular flows, the concept emerged to combine two of them into a single framework. In this research, the aim is to integrate a precise collision model based on DEM as described in [135] with a straightforward HEL method.

The current study presents a streamlined collision model created within the context of machine learning (ML), which can be seamlessly incorporated into the conventional HEL approach, replacing the KTGF closure components. The resultant model decreases computational requirements, while upholding the anticipated level of precision. In last few years, and still today, ML gains more and more interest, as the tool to model or optimize complex processes [145, 16, 90], or refining models through data-analysis [142], and predicting and confirming signals [91]. Beyond its notable accuracy, the developed simplified model also significantly trims computational time. For instance, the reduced order model (ROM) technique has been employed to simulate furnace operations under flameless conditions, aligning with the concept of a digital twin, as described in [27]. These mentioned studies effectively showcase the practicality of ML algorithms as potent tools, holding substantial potential for curtailing the expenses associated with numerical simulations.

In the first part of the work a surrogate collision model (SCM), created using, a combination of available CFD approaches with ML algorithm. To be more precise, in this part algorithms based on decision trees will be used. ML model will be built, using the data gathered using DEM approach. To be more precise, the SCM was integrated into the standard HEL technique, which is available in ANSYS®Fluent software, replacing KTGF with a set of user-defined functions (UDFs). A comparison was made between the outcomes of the Hybrid Euler-Lagrange Surrogate Collision Model (HELSCM) and the

numerical results derived from both the standard HEL and DEM methods, as well as the experimental measurements reflecting conditions in simulations. For the purpose of comparing the experimental with simulation results, an in-house algorithm was created for purposes of post-processing measurement data in LabView.

In the second part of the dissertation, the machine learning tools were also used to predict particle interactions influencing their distribution in a boiler, but at a different level and in a different way. Developed ML model was used to predict field values of the pressures, velocities of gases and solids, based on high-level of CFD simulation applying complex HEL approach. The main idea of this work, was to integrate hybrid modeling system with the Łagisza Power Plant CFB's 460 MWe boiler control system. Such integration aims to establish an enhanced intelligent control setup, focusing on reducing emissions and enhancing overall performance and support prediction and perspective algorithms for boiler control and operation optimization. The main research question concerns the development of a modeling system, that simulates the CFB furnace in sufficient detail and speed, for use as a prediction and prescription system as mentioned. This part of the work presents the development and validation of the hybrid modeling approach, that integrates various modeling techniques. The approach uses neural networks to improve the prediction of field values, based on collected results from digital twin model. It makes it possible to bypass, performing full CFB calculations for the boiler, and thus significantly speed up to obtain results that can be used in on-fly mode for boiler control. It is worth to mention here that developed here methodology has been incorporated in OPTI_AI_UNIT (08/060/FSB/1004) project.

1.4. Current status of the research

Numerous methods are presently being used for the modeling of multiphase flows. It includes approaches, already existing for an extended period. It is important to acknowledge that they were effectively utilized in numerous simulations. Additionally, there are also emerging techniques that are unveiling their promise and are currently undergoing developmental phases. Above mentioned approaches, which are included in the current work, are utilized to model granular flows, characterized by high solid volume fraction, where the number of interactions between particles is very large due to their number. There are few commonly used commercial CFD software packages, such as Fluent [34], ANSYS CFX [2], STAR-CD [5], which allows to introduce users changes to the source code by externally written functions.

The standard EE approach was applied to model flows in a pipe in a thin bent pipe, where suspended, plug and dense flows are distinguished [87]. Another paper presents simulations of circulating fluidized bed in real scale boiler [9]. The EE approach was also

used for simulations of the dilute phase pneumatic conveying in horizontal pipe [25]. In that work, the dilute flow was analized, but even in such case only the monodispersed particles were investigated instead of polidispersed. Another work presents usage of EE method to simulate the dilute flow of glass particles, in a bend pipe [66].

Another work presents the usage of EE and HEL approach for the erosion detection and influence. The mixing of approaches was based on assumption of using EE approach in the subdomains, where the effect of particles on erosion was not that significant. In the subdomains the most vulnerable to erosion the HEL approach was applied. It is worth noting that this work only considered the effect of fluid on particles, but not particles on fluid [88]. The same simulations were performed using only HEL model, and the simulation times were compared. As the result model connection showed the decrease in time and computational effort. The another article presents the comparison of HEL and multiphase particle-in-cell (MP-PIC) technique, also in case of modeling fluidized bed [31]. The HEL was validated, for the comparison of the results in case of particle collisions in configuration one to one, and when they were grouped in parcels, for simplification.

The work of [144] presents the combination of of the open source code OpenFOAM [7] with the PFC^{3D} (Particle Flow Code). It investigates the characteristics of the particles flow in a pipe. Proposed model serves as valuable benchmark for ex. solving the soil erosion. Barasso et al. [28] proposed the combination of popoulation balance models (PBM) with DEM results, using ROM models, built using Artificial Neural Networks (ANN). Such combination, led to succesfully decrease the simulation time while keeping the accuracy of DEM model. Interactions of particles in the rotating drums, are subjected to careful analysis, because, as in the case of fluidized beds, they are characterized by the multiplicity and complexity of collisions. Here, too, the classical DEM approach is too time-consuming, so the work of Y. Li et al.[84] presents the use of Support Vector Machine for Regression (SVR). SVR model predicts the angle of response, as well as energy of collision. As variables determining the prediction result, the model takes into account drum size, rotation speed, filling level. The accuracy of the model's performance was determined using coefficient of determination, where it was equal to 0.92 for the angle and 0.86 for the collision energy. The paper also mentions that the model can be used for other devices, where there is a multiplicity of collisions between particles. The analyze of the movement of particles of different diameters under vibration is discussed in [24]. Here, instead of using the traditional DEM approach, machine learning models were used to predict particle relocation. The influence of varying vibration amplitudes frequencies on the percolation of particles, each with distinct size ratios, were examined concerning both percolation velocity and radial dispersion.

Another article utilizes ROCKY-DEM for the rotating drum particle simulations [92]. One-diameter particles got under consideration, whose simulations of mixing in the drum showed agreement with experimental results. There is potential to extend the simulations to different diameters. X. Ma et al. [86] took under consideration the motion of a single particle in a rotating flow field. It is important issue, to understand the exact interaction and behaviour of a single particle. Simulations were performed in EDEM software, which is part of ANSYS®Fluent. The results of the simulations were compared with the experimental results. The work, however, does not deal with inter-particle interactions, a step that is yet to be taken. It focuses on the behavior of only one particle in a rotating field is considered. Another proposal is the use of EDEM and LIGGGHTS [78] software to simulate the charging system of a blast furnace for iron production [136]. The charging system is designed to deliver the pellets to downstream parts of the system, through a so-called chute. Simulation results from both approaches are compared with experimental results. Both showed agreement in the results with the experimental measurements. The use of LIGGGHTS software significantly reducing calculation time.

Article [41] presents the use of ANN to model CFB flow, focusing on local voidage fluctuations being the part of a dynamic bed. To gather the data of the variations in local voidage, for training of ANN, an optical transmittance probe at different axial and radial locations. By utilizing this model, it became feasible to make forecasts for both short-term and long-term dynamic features within the circulating fluidized bed. The proposed approach by M. Nadda et al.[96] describes the use of results obtained from CFD simulations of fluidized bed behavior, to train ANN to reliably forecast a wide range of hydrodynamic characteristics within fluidized beds. The accuracy of the model was determined using the mean squared error (MAE) between the prediction and the expected value, derived from CFD simulations. Its value is in the range of 1.61- 5.5%. The proposed approach is similar to the one described in this paper, although it employs a 2D modeling approach.

1.5. Outline of the thesis

The Ph. D. thesis is divided into 7 main chapters. The dissertation is focused around the topic of interparticle interactions, as a crucial aspect in the context of fluidization processes.

The *introductory* chapter discusses the core aspect of the subject it is fludization phenomena. The description of the technology itself is included, together with the bubbling and circulating fluidized bed specifically. Additionally, the particle hydrodynamic is introduced. This chapter presents valuable terminology and introduces references to other relevant works, as well as the objectives of the dissertation.

The *mathematical models* chapter is devoted to provide mathematical description of the computer models used for modeling multiphase flows. The primary emphasis lies in the models employed to simulate granular flow phenomena, as detailed in the paper.

The *machine learning and neural networks* chapter is dedicated to the machine learning models used in the work, and a description of neural networks, which are part of a group of artificial intelligence belonging to the machine learning family. In addition to a general description of the issue, a mathematical description of the methods used is also presented.

The *experimental test-rig for tracking particle collisions* chapter describes experimental work, numerical model validation. Also it presents development of surrogate model in terms of ML technique based on the DEM data, as well as the comparison of the simulation results against the experimental measurements.

The *development of simplified collision approach* chapter discusses the surrogate model in terms of machine learning technique based on the DEM data. It covers the stage of data preparation, with the description of the approach. Also, the model building itself as well as its implementation to commercial software is described. The comparison of the simulation results against the experimental measurements are included in the chapter. In addition, simulations of the fluidization phenomenon in a low-scale boiler are described, allowing verification of the use of a simplified model to reflect this phenomenon.

The *hybrid approach for modeling large scale CFD boiler* chapter is devoted for application of deep neural networks, as a method to support a modeling framework capable of accurately representing and swiftly simulating the CFB furnace, designed for utilization as a predictive and prescriptive system.

Finally, the last chapter *discussion and conclusions* concludes the thesis with summary of the achievements and recommendations for future research.

2
Mathematical models

Modeling of the granular flows, e.g. fluidization, is one of the most challenging due to the large number of particles, resulting in a large number of interactions between them. An additional factor increasing the difficulty, is the size of the devices, in which the phenomena are modeled. It directly affects the number of particles to be analyzed, as well as introducing complications in the determination of fluid interactions.

The chapter provides a mathematical description of the available computational models so that the next chapter can propose a surrogate approach based on combination of available models. The main goal will be to speed up simulations, while maintaining the accuracy of the results. These two factors are highly desirable, given the need to test new technological solutions. The aim of this work is, to propose an alternative, or to make small improvements to already existing and successfully applied computer methods, by combining them with machine learning or artificial neural network algorithms.

2.1. Standard Euler-Euler approach

The most popular approach used to model granular flow is the multi fluid Euler-Euler (EE), which found a lot of practical applications [112, 21]. This model is very stable, and all his features are well recognized and studied. To describe the transport of the particles in an isothermal flow, i.e. without mass exchange, in the EE approach, a set of equations, including the conservation of mass and momentum are used. In this technique, the gas, liquid and dispersed phases are treated as interpenetrating continua. Therefore, the dispersed phase, is treated as it was possessing characteristics fluid properties.

The transport equations for each individual component are provided in an instantaneous format, excluding source part. Interested readers are referred to [57, 58, 70] for details, about derivation of the below equations. Starting with mass conservation equation for the continuous phase

$$\frac{\partial}{\partial t}\left(\varepsilon_f \rho_f\right) + \nabla \cdot \left(\varepsilon_f \rho_f \mathbf{u}_f\right) = 0 \tag{2.1}$$

The mass conservation equation for the dispersed phase

$$\frac{\partial}{\partial t}\left(\varepsilon_s \rho_s\right) + \nabla \cdot \left(\varepsilon_s \rho_s \mathbf{u}_s\right) = 0 \tag{2.2}$$

The equation of conservation of momentum for the continuous phase

$$\frac{\partial}{\partial t}\left(\varepsilon_f \rho_f \mathbf{u}_f\right) + \nabla \cdot \left(\varepsilon_f \rho_f \mathbf{u}_f \mathbf{u}_f\right) = -\varepsilon_f \nabla p + \nabla \cdot \tau_f + \varepsilon_f \rho_f \mathbf{g} + \sum_q^{N_s} \left[K_{qf}\left(\mathbf{u}_f - \mathbf{u}_q\right)\right] \quad (2.3)$$

The equation of conservation of momentum for the dispersed phase

$$\frac{\partial}{\partial t}\left(\varepsilon_s \rho_s \mathbf{u}_s\right) + \nabla \cdot \left(\varepsilon_s \rho_s \mathbf{u}_s \mathbf{u}_s\right) = -\varepsilon_s \nabla p + \nabla \cdot \sigma_s + \varepsilon_s \rho_s \mathbf{g} + K_{fs}\left(\mathbf{u}_f - \mathbf{u}_s\right) + \sum_q^{N_s - 1} \left[K_{qs}\left(\mathbf{u}_q - \mathbf{u}_s\right)\right] \quad (2.4)$$

where $\mathbf{g}$ is the gravity, the subscripts f and s denote the gaseous and solid phases respectively, ε denotes the phase volume fraction, ρ is the density, $\mathbf{u}$ defines the velocity vector, p is the pressure of gaseous phase, and K represents the interphase exchange coefficients between phases, the subscript q stands for the q-th solid phase, N_s is the total number of solid phases. The τ_f is the fluid stress tensor which represents viscous forces. The fluid stress tensor can be calculated as [3]

$$\tau_f = \varepsilon_f \mu_f \left(\nabla \mathbf{u}_f + \nabla \mathbf{u}_f^T\right) + \varepsilon_f \left(\lambda_f - \frac{2}{3}\mu_f\right) \nabla \cdot \mathbf{u}_f \bar{\mathbf{I}} \quad (2.5)$$

where λ_f is the bulk viscosity of fluid phase, $\bar{\mathbf{I}}$ is the unit tensor, μ_f represents the fluid dynamic viscosity. In case of an incompressible flow, the term $\nabla \mathbf{u}_f$ as well as the bulk viscosity in Eq. 2.5 are equal to zero. To predict field variables, quantifying by the average volume fraction occupied by each phase, the group of multiphase transport equations are solved in Reynolds averaged form [47].

To calculate the solid stress tensor, denoted as σ_s, which represents particles interactions, it is necessary to determine several closure terms. Such terms are used to define the granular pressure, solid bulk viscosity and shear viscosities. The solid stress tensor, is determined as [124]

$$\sigma_s = -p_s \bar{\mathbf{I}} + \varepsilon_s \mu_s \left(\nabla \mathbf{u}_s + \nabla \mathbf{u}_s^T\right) + \varepsilon_s \left(\lambda_s - \frac{2}{3}\mu_s\right) \nabla \cdot \mathbf{u}_s \bar{\mathbf{I}} \quad (2.6)$$

where λ_s is the bulk viscosity, $\bar{\mathbf{I}}$ is the unit tensor, p_s is the granular pressure, μ_s stands for the solid dynamic viscosity, and $\mathbf{u}_s$ for the average velocity vector of the solid phase obtained at the particle location. The next step, in modeling granular phase is the mathematical description of the dynamic and bulk viscosity, as well as the solid pressure. The dispersed phase modelling, also understood as the momentum exchange occurring between the solid phase, at different solid volume fractions, determines the flow regime in the system. If, in the system, the solid volume fraction is low (less than 61 % [58]), the particle collisions and kinetic transport are significant. Otherwise, when the solid volume fraction is high, the particles are in contact, thus the collisions number decreases. As

the effect, the transport is controlled by the friction between particles and the kinetic transport. The dynamic viscosity of the solid phase, can be expressed as the sum of three terms

$$\mu_s = \mu_{s,\text{kin}} + \mu_{s,\text{col}} + \mu_{s,\text{fric}} \tag{2.7}$$

where $\mu_{s,\text{col}}$, $\mu_{s,\text{kin}}$, and $\mu_{s,\text{fric}}$ are the viscosity due to collisions, kinetic transport and friction, respectively.

The correlations, between the viscosities resulting from kinetic transport [58] and collisions [124] are applied

$$\mu_{s,\text{kin}} = \frac{\varepsilon_s \rho_s d_s \sqrt{\Theta\pi}}{6(2-e_{ss})}\left[1+\frac{2}{5}(1+e_{ss})(3e_{ss}-1)\varepsilon_s g_{0,ss}\right] \tag{2.8}$$

$$\mu_{s,\text{col}} = \frac{4}{5}\varepsilon_s^2 \rho_s d_s g_{0,ss}(1+e_{ss})\left(\frac{\Theta}{\pi}\right)^{1/2} \tag{2.9}$$

where e_{ss} stands for the inelasticity of particle collisions, also defined as the restitution coefficient, and Θ is the granular temperature. The probability of particle collisions $g_{0,ss}$ can be determined as [102]

$$g_{0,ss} = \left[1-\left(\frac{\varepsilon_s}{\varepsilon_s^*}\right)^{1/3}\right]^{-1} \tag{2.10}$$

where ε_s^* represents the permissible limit of particle packing in the control volume, equals to 63 % by volume of the solid phase [58]. While, the solid volume fraction exceeds the defined friction limit value (equal to 0.61 %) ε_s^{fr}, the model for dense regime (friction regime) is activated. In such case, the term responsible for determination of the collisions, in Eq. 2.7, aims for zero while taking into account friction $\mu_{s,\text{fric}}$, defined as

$$\mu_{s,\text{fric}} = \frac{p_s \sin\phi}{2\sqrt{I_{2D}}} \tag{2.11}$$

where I_{2D} is the invariant of the stress tensor, ϕ defines the angle of alignment of particles, p_s stands for the granular pressure. The granular pressure, likewise granular viscosity, occurs as the sum of three terms, representing kinetic transport, collisions and friction. The relationship for granular pressure, in the case of dilute flows, with a volume fraction below the friction limit (61%), can be written as

$$p_s = \underbrace{\varepsilon_s \rho_s \Theta}_{p_{s,\text{kin}}} + \underbrace{2\varepsilon_s^2 \rho_s (1+e_{ss}) g_{0,ss} \Theta}_{p_{s,\text{col}}} + p_{s,\text{fric}} \tag{2.12}$$

where the first term stands for kinetic transport, and the second defines the value of the pressure, resulting from the collisions. The last term, is accounted for the dense flow,

where ε_s, takes value higher than 61 % [58, 85]. In the dense regions, where the solid volume fraction exceeds the solif volume fraction defined at the friction limit ε_s^{fr}, the collision term, as for granular viscosity, tends towards zero and the friction modelling term becomes dominant $p_{s,fric}$. The term is determied as [38, 123]

$$p_{s,\text{fric}} = 0.1\varepsilon_s \frac{\left(\varepsilon_s - \varepsilon_{s,min}\right)^2}{\left(\varepsilon_{s,max} - \varepsilon_s\right)^5} \tag{2.13}$$

The bulk viscosity, appearing in Eq. 2.6, describes the resistance of the solid phase to a change in volume [85]

$$\lambda_s = \frac{4}{3}\varepsilon_s \rho_s d_s g_{0,ss}(1+e_{ss})\sqrt{\frac{\Theta}{\pi}} \tag{2.14}$$

The granular temperature cannot be unambiguously measured, as no specific thermometer is known for such purpose. It can understood as kind of turbulent kinetic energy or the energy of solid fluctuations, and be described as mean square value of the random particle velocity fluctuations about the mean flow velocity [60]. A brief description of characteristics concerning granular temperature, can be found in the research [61]. Granular temperature Θ, present in the equations describing the granular pressure, viscosity and drag force represents fluctuations of the velocity of the particles C

$$\Theta = \frac{1}{3}\left\langle C_x^2 + C_y^2 + C_z^2 \right\rangle \tag{2.15}$$

where the granular temperature is determined each solid phase s as

$$\frac{3}{2}\left[\frac{\partial}{\partial t}(\varepsilon_s \rho_s \Theta) + \nabla \cdot (\varepsilon_s \rho_s \Theta \mathbf{u}_s)\right] = \nabla \cdot (k_\Theta \nabla \Theta) - \nabla p_s + \tau_s : \nabla \mathbf{u}_s - \gamma_s + \varphi_{fs} \tag{2.16}$$

where γ_s is a dissipative term representing the rate of energy dissipation within the solid phase due to particle interactions, $\tau_s : \nabla \bar{u}_s$ is the fluctuating energy due to the forces between particles (viscous dissipation), $\nabla \cdot (k_\Theta \nabla \Theta)$ represents the diffusion, φ_{fs} is the exchange term representing the kinetic energy transfer between phases [124], k_Θ is the conductivity of the granular temperature and p_s is the granular pressure. Due to high instabilities in Eq. (2.16) the simplification was applied by omiiting the convection term. As the result the differential equations are transformed to algebraic form [131, 124]. The differential equation for the granular temperature is then simplified to

$$-\nabla p_s + \tau_s : \nabla \mathbf{u}_s - \gamma_s + \varphi_{fs} = 0 \tag{2.17}$$

The energy dissipation from the particle interactions γ_s is calculated using the formula proposed by Lun at al. [85]

$$\gamma_s = \frac{12(1-e_{ss}^2)g_{0,ss}}{\bar{d}_s\sqrt{\pi}}\rho_s\Theta^{3/2}\varepsilon_s^2 \tag{2.18}$$

where $g_{0,ss}$ is the radial distribution function, $\bar{d}_s$ is the average particle size in numerical cell. If the restitution coefficient e_{ss} in Eq. (2.18) attends to unity, the dissipation of kinetic energy within the solid phase becomes negligible $\gamma_s \to 0$.

One of the key parameters, responsible for the interactions between particles and the fluid, is the drag exchange coefficient K between phases. The relationship, specified by the drag coefficient is most frequently obtained experimentally, based on the pressure drop in fluidized bed or settling tank. In the current work, the model described by was used [58]. Such model, combines two separate into one: Ergun's model, describing flow in the packed bed (approximately where volume fraction is above 20%), and the Wen&Yu model [137] dedicated for flows, where volume fraction is below 20%. Combinations of Ergun and Wen&Yu models, are described using Eqs. 2.19 and 2.20 respectively

$$K_{\text{Ergun}} = 150\frac{\varepsilon_s^2\mu_f}{\varepsilon_s d_s^2} + 1.75\frac{\rho_f\varepsilon_s\left|\mathbf{u}_s-\mathbf{u}_f\right|}{d_s} \tag{2.19}$$

$$K_{\text{Wen\&Yu}} = 0.75C_{\text{D}}\frac{\varepsilon_s\varepsilon_f\rho_f\left|\mathbf{u}_s-\mathbf{u}_f\right|}{d_s}\varepsilon_f^{-2.65} \tag{2.20}$$

where C_{D} is the drag coefficient defined as

$$C_D = \frac{24}{\varepsilon_f\text{Re}}\left[1+0.15(\varepsilon_f\text{Re})^{0.687}\right] \tag{2.21}$$

Above models, do not take into account the situation, where volume fraction is equal to 20% [68]. To overcome such discontinuity, often a blending function is applied. The drag coefficient is defined as [68]

$$K = \phi(\varepsilon_s)K_{\text{Ergun}} + (1-\phi(\varepsilon_s))\,K_{\text{Wen\&Yu}} \tag{2.22}$$

where $\phi(\varepsilon_s)$ is a blending function written as [146]:

$$\phi(\varepsilon_s) = 0.5 + \frac{\arctan\left[262,5(\varepsilon_s-0,2)\right]}{\pi} \tag{2.23}$$

If two or more additional diffuse phases are modelled, an additional drag coefficient between these phases must be calculated. For this purpose, the interphase symmetrical drag coefficient proposed by Syamlal et al. [124] should be used

$$K_{\text{sym},qs} = \frac{3(1+e_{qs})\left(\frac{\pi}{2}+C_{\text{fric},qs}\frac{\pi^2}{8}\varepsilon_s\rho_s\varepsilon_q\rho_q\left(d_q+d_s\right)^2 g_{0,qs}\right)}{2\pi\left(\rho_q d_q^3+\rho_s d_s^3\right)}\left|\mathbf{u}_q-\mathbf{u}_s\right| \tag{2.24}$$

where q stands for an additional dispersed phase, e_{qs} is the coefficient of restitution, $C_{\text{fric},qs}$ is the the friction between the q th and s th solid phase particles and $g_{0,qs}$ is the radial distribution function.

2.2. Hybrid Euler-Lagrange approach

The main bottleneck of the EE technique is a problem with modeling a real particle size distribution (PSD) of the fluidization material. Most of applications uses only two or three additional solid phase to represents the real PSD. Even in such configuration, the number of transport equation needed to be resolved can easily exceed twenty. To resolve this problem, a Lagrangian technique was introduced [65, 30].

A commonly used alternative to the Euler-Euler is the Hybrid Euler-Lagrange approach. Such approach is described by few mathematical models. The most frequently used are the Dense Discrete Phase Model (DDPM) [3, 45] ot the Multiphase Particle in Cell (MP-PIC) model [23, 108, 122]. The mutual basis of both models, mentioned above, is the four-way coupling procedure [94] considering the interaction between phases, and interactions occurring in the dispersed phase. In the HEL technique, collisions are modeled in the basis of the Kinetic Theory of Granular Flow (KTGF) [69, 46, 58, 85], which takes into account interactions based on primary interpretation, developed for kinetic molecular theory of dense gases [38]. In the continuous phase, the influence of particle movement and the energy exchange between them, are accounted by two additional and the fluid phase, are included in terms in the conservation equations. The dispersed phase is treated both, as continuum and as dispersed phase, where particles are tracked in Lagrangian frame of reference. The particle properties, are interpolated to and from the Eulerian grid. The interpolation operators [23, 120], transfer data between particle positions and Eulerian grid. The interpolation method is used to resolve the stresses between particles in the dispersed phase, which are difficult to calculate for each particle separately. It is presumed, that the gradient of the stress, occurring in the dispersed phase, can be determined on the Eulerian grid, and interpolate its value to the discrete particle position [122]. In case of HEL approach, for simplification of calculations, the motion equation is solved for parcels (containing multiple particles), instead of individual particles. Such particle parcels have the same physical properties as density, mass, position, composition, etc. The number of particles contained in a single parcel is determined from

$$n_p = \frac{\dot{m}_{\text{parcel}} \Delta t}{m_p} \tag{2.25}$$

where Δt is the time step in transient calculation, $\dot{m}_{\text{parcel}}$ mass flow rate of single parcel and m_p is the mass of individual particle evaluated based on the particle diameter and

density. The equation 2.26 determines the motion of a particle, resulting from the forces acting on it [123]

$$\frac{\mathrm{d}\mathbf{u}_p}{\mathrm{d}t} = F_{\mathrm{D}}(\mathbf{u}_f - \mathbf{u}_p) + \frac{\mathbf{g}(\rho_p - \rho_f)}{\rho_p} - \frac{\nabla p}{\rho_p} - \frac{\nabla \cdot \sigma_s}{\rho_p} \tag{2.26}$$

where the subscript p denotes the particle data (using this approach, it is possible to simulate different materials of patricles within separate dispersed phases). σ_s stands for the granular stress tensor, which represents particles interactions calculated based on the KTGF [58, 38, 85] in the Eulerian grid. ρ_p is the particle material density, $F_{\mathrm{D}}(\mathbf{u}_f - \mathbf{u}_s)$ is the acceleration due to the drag. The term $\frac{\nabla \sigma_s}{\rho_p}$ is the velocity and position of particle, determined based on solid volume fraction in computational cell. $-\nabla p/\rho_p$ is the particle acceleration, resulting from pressure difference at the particle location. Having the location of a particle, its velocity can be determined by a relationship

$$\frac{\mathrm{d}\mathbf{x}_p}{\mathrm{d}t} = \mathbf{u}_p \tag{2.27}$$

where x_p is the particle position. Stepping further, the solid volume fraction in certain cell, where the particle is positioned is calculated as

$$\varepsilon_s = \frac{\sum_{i=1}^{N_{\mathrm{parcel}}} V_{p,i} n_{\mathrm{p},i}}{V_{\mathrm{c}}} \tag{2.28}$$

where V_p and V_{c} are the volume of parcel and cell respectively. The volumetric share is calculated and assigned to the analyzed cell, where the gas fraction is determined as $\varepsilon_g = 1 - \varepsilon_s$. By utilizing the KTGF and having determined the volume ratio between gases and solids, one can establish the parameters required to compute the stresses, determined by the formula 2.6.

Utilizing Hybrid Euler-Lagrange approach, similarly as in the Euler-Euler the equations of mass, momentum and energy conservation are solved. For the first mention, the equation follows as [45]:

$$\frac{\partial}{\partial t}(\varepsilon_f \rho_f) + \nabla \cdot (\varepsilon_f \rho_f \mathbf{u}_f) = S_{\mathrm{DDPM_m}} \tag{2.29}$$

The momentum conservation equation is determined as

$$\begin{aligned} \frac{\partial}{\partial t}(\varepsilon_f \rho_f \mathbf{u}_f) + \nabla \cdot (\varepsilon_f \rho_f \mathbf{u}_f \mathbf{u}_f) = -\varepsilon_f \nabla p + \\ \nabla \cdot \tau_f + + \varepsilon_f \rho_f \mathbf{f} + K_{sf}(\mathbf{u}_s - \mathbf{u}_f) + S_{\mathrm{DDPM_u}} \end{aligned} \tag{2.30}$$

where $\mathbf{g}$ is the gravity, the subscripts f and s denote the gaseous and solid phases respectively, ε denotes the phase volume fraction, ρ is the density, $\mathbf{u}$ defines the velocity

vector, p is the pressure of gaseous phase, and K represents the interphase exchange coefficients between phases, the subscript q stands for the q-th solid phase, N_s is the total number of solid phases. The τ_f is the fluid stress tensor which represents viscous forces, S_{DDPM_m} denotes the mass exchange, and S_{DDPM_u} defines momentum exchange.

2.3. Discrete Element Method

Using above mentioned approaches, a user might encounter stabilization problems, mass conservation problems and more briefly described in [55, 13]. Different model, offering possibility to simulate granular flows is Discrete Element Method (DEM). There is no change in solving the fluid flow, which is solved in Eulerian grid, and the particles are tracked in the Lagrangian frame of reference (see equations 2.29 and 2.30). In contrary to HEL approach, it allows collision detection, based on the common space position of colliding particles and their velocity vectors [128]. However, at the same time, such approach makes calculations drastically more expensive. Taking into account the change in the way of collision detection, using the DEM model, the equation of motion of the particle differs from the 2.26 equation. An additional terms appear, as well as the term responsible for determining the force resulting from a collision between particles. After modifications, the particle position is determined as [146]

$$\frac{\mathrm{d}\mathbf{u}_p}{\mathrm{d}\tau} = F_{\mathrm{D}}(\mathbf{u}_f - \mathbf{u}_p) + \frac{\mathbf{g}(\rho_p - \rho_f)}{\rho_f} + F_{\mathrm{other}} + F_{\mathrm{col}} + F_{\mathrm{fric}} \tag{2.31}$$

where F_{fric} denotes the forces occurring between the particles due to friction, which are influenced by the material properties of the particles, F_{other} are the additional forces e.g. rotational force. In case of DEM, two sub-models can be chosen for definition of the collision forces acting on the interacting particles i.e. soft-sphere and hard-sphere [47]. In case of soft-sphere approach the deformations resulting from collisions are taken into account. A visualization of such a collision, with its exaggerated effect, is presented on the Fig. 2.1.

The force resulting from a collision is determined as [146]

$$\mathbf{F}_{\mathrm{col}} = \mathbf{F}_1 = -\mathbf{F}_2 = (K\delta + \gamma(\mathbf{u}_{p,12}\overline{\mathbf{e}}_{\mathbf{12}}))\overline{\mathbf{e}}_{\mathbf{12}} \tag{2.32}$$

where K is the spring constant, δ is the deformation, γ is the damping coefficient, $\mathbf{u}_{p,12}$ defines relative velocity $(\mathbf{v}_2 - \mathbf{v}_1)$ between particles, $\overline{\mathbf{e}}_{\mathbf{12}}$ is the unit vector between particle 1 to 2, calculated as

$$\overline{\mathbf{e}}_{\mathbf{12}} = \frac{\mathbf{x}_2 - \mathbf{x}_1}{||\mathbf{x}_2 - \mathbf{x}_1||} \tag{2.33}$$

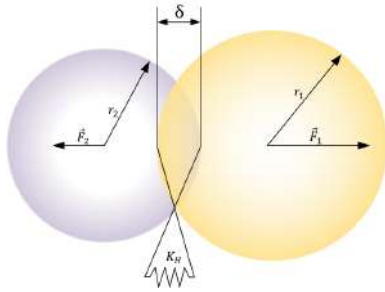

Fig. 2.1. Exaggerated moment of collision of two particles, with marked occurring forces

where x_1 and x_1 are the positions of the particle 1 and 2 respectively. The damping coefficient, included in the equation 2.32 is determined as

$$\gamma = -2\frac{m_{12}\ln\eta}{\tau_{col}} \tag{2.34}$$

where τ_{col} stands for the collision time, defined as $f_{loss}\sqrt{\frac{m_{12}}{K}}$ and f_{loss} is determined from $\sqrt{\pi^2 + \ln^2\eta}$. The restitution coefficient takes value from 0 to 1, and influences the amount of energy dissipated due to collision, both between particles and particles-wall. For η equal to 0, the collision energy is fully dissipated, while for η equal to 1 the the impact is fully elastic. The term, so-called 'reduced mass' is determined as

$$m_{12} = \frac{m_1 m_2}{m_1 + m_2} \tag{2.35}$$

where m_1 and m_2 is the mass of particle 1 and 2 respectively. Additional force, resulting from the friction between two colliding bodies is defined as [146]

$$F_{fric} = \mu_{fric} F_n \tag{2.36}$$

where μ_t stands for friction coefficient, and F_n is the magnitude of the normal to the surface force.

In case of using non-linear Hertzian sub-model, the collision force due to particle contact is calculates as

$$\mathbf{F}_1 = \mathbf{F}_2 = K_H \delta^{3/2} \overline{\mathbf{e}}_{\mathbf{12}} \tag{2.37}$$

where K_H is the physical material property, determined from the respective Young's Moduli E_1 and E_2 of the two colliding particles and their Poisson's ratios ν_1 and ν_2

$$\mathbf{K_H} = \frac{4}{3}\frac{E_1 E_2}{E_2(1-\nu_1^2)E_1(1-\nu_2^2)}\sqrt{\frac{r_1 r_2}{r_1 + r_2}} \tag{2.38}$$

For more, detailed information and more possible configurations of the discrete model DEM, the interested readers, are reffered to [129, 128, 143, 47]. When using the DEM model, it is necessary to use a sufficiently small particle tracking time, which will not allow a non-naturally large deformation of the particles due to the collision. The exact methodology for selecting the appropriate value is briefly described in [148]. The final time step for tracking particle parcel is selected based on relation

$$t = 0.2 \cdot \min\left(t_{\text{Raleigh}}, t_{\text{Hertz}}, t_{\text{Cundall}}\right) \tag{2.39}$$

The Raleigh (Eq. 2.40) and Hertz (Eq. 2.41) time equation are calculated as [78]

$$t_{\text{Raleigh}} = \frac{\pi r_{sphere}}{0.1631(\nu + 0.8766)} \sqrt{\frac{\rho_p}{E/[2(1+\nu)]}} \tag{2.40}$$

$$t_{\text{Hertz}} = 2.87 \left[\frac{(\rho_p (4/3) \pi {r_{sphere}}^3)^2}{r_{sphere} E^2 V_{max}} \right]^{0.2} \tag{2.41}$$

For the Cundall model, used in the software LS-DYNA DEM [132], the time step is determined as

$$t_{\text{Cundall}} = \sqrt{\frac{\rho_p (4/3) \pi {r_{sphere}}^2)}{\frac{E}{[3(1+\nu)]} NORMK}} \tag{2.42}$$

where $NORMK$ is a stiffness penalty parameter, typically ranging between 0.1 and 0.001.

3
Machine learning and neural networks

Both Machine Learning and Neural Network Models branch into two subtypes, ie. supervised and unsupervised learning. In the first case, such models require human to prepare the data accordingly beforehand. Such data are subjected to careful analysis. Often, especially in the case of classification problems, pre-labeling of data is required. In case of supervised learning, for the inputs a corresponding outputs must be assigned. Such approach is widely used in various applications, such as image recognition [133], natural language processing [83], spam detection [50], medical diagnosis [141, 118, 37], and many other tasks where labeled data is available and predictions need to be made based on that data. Widely used models for such purpose are ex. Linear Regression, Decision Trees, Neural Networks, Random Forests, Ensembled Methods. In such work, last ones will be described more briefly in next sections. On the other hand, unsupervised learning, does not require humans interference. The algorithms, explores the data on its own, seeking for patterns, structures within the data without any predefined guidance or target variable. Due to an algorithm analyzing the data, it is possible to find hidden connections between the data, not necessarily observable by humans. As only supervised models were utilized in this study, no detailed explanation of any model from the second category is provided. Depending on the problem to be solved, for both of the above groups, both of these groups are categorized into either regression or classification. In case of classification problem, the target values are usually integers, referring to a class. In the case of a single-class problem, where the outputs are denoted as, for example, true/false, we are dealing with denotations after preprocessing, most often the pair of 0 and 1. However, also occurring cases are multi-class problems, where there are more possibilities than true/false, therefore, the target integers are more diversified. For regression problems, the output or the target variable is a continuous value, most frequently a float.

3.1. Decision trees

Decision Trees (DTs) are a part of a supervised learning method groups, used for both classification and regression problems. As for all machine learning models, the goal is to predict the target variable, based on some inputs, by learning simple decision rules. Speaking further about decision trees, it is necessary to introduce some concepts related

to them, concerning the processes occurring during learning, and the names of the various elements of the tree.

- Root Node - represents the entire population or sample and this further gets divided into two or more homogeneous sets,
- Splitting - a process of dividing a node into two or more sub-nodes,
- Decision Node - a split from previous node,
- Leaf / Terminal Node - the node without any further splits,
- Pruning - removal of sub-nodes of a decision node, is called pruning. In different words it is an opposite process of splitting,
- Branch / Sub-Tree - a subsection of the entire tree,
- Parent and Child Node - a node, divided into sub-nodes is called a parent node of sub-nodes whereas sub-nodes are the child of a parent node.

In reference to the name of the method itself, decision trees (both classification and regression) are built based on a tree structure. The dataset brakes down into smaller and smaller subsets, which in turn leads to an increase in the size of the tree itself, so the development proceeds.The final result is a tree with decision nodes and leaf nodes. A diagram of the resulting tree is presented in Fig. 3.1, along with a description of the various elements.

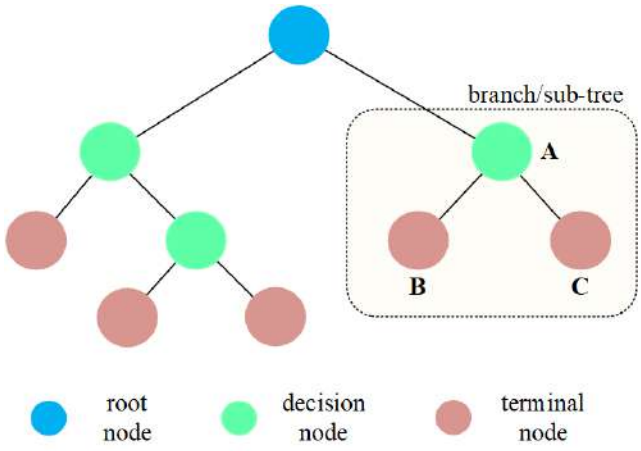

Fig. 3.1. Scheme of simple decision tree, with marked main elements, as well as highlited branch, parent and child nodes [72]

In general there are two types of decision trees with the categorical and continuous output. The principle of operation is similar, but the difference lies in determining the division of the data into smaller groups at the time of distribution to subsequent decision nodes.

3.1.1. Gini Impiurity

Proposed by [35], the gini index is the measure of non-homogenity. It is very popular, and easiest method, used for classification trees, for splitting the nodes. It is calculated by subtracting the sum of the squared probabilities of each class from one, the formula is as follows

$$GiniIndex = 1 - \sum_{i=1}^{n} p_i^2 \tag{3.1}$$

where p_i is the probability of the class i. In general, in case of a pure node, the Gini Index would be equal to 0.

3.1.2. Enthropy

Entropy [117] is a measure of the orderliness of a system, or information. The higher its value, the harder it is to find relationships between the data present. The formula for an enthropy is

$$Entrophy = -\sum_{i=1}^{n} p_i log_2 p_i \tag{3.2}$$

To determine the root node, the entrophy is calculated for each variable and its potential splits. Having this value, it is possible to determine a metric to measure the reduction of a disorder in the target class. For such purpose the Information Gain is calculated, using the formula:

$$InformationGain = Enthropy_{parent} - Enthropy_{children} \tag{3.3}$$

where index *parent* reflects to the parent node, while *children* represents the average entropy of the child nodes.

3.1.3. Reduction in variance

Reduction in variance [110] is used to solve problems with, so called, continuous target. It is also used to calculate the homogeneity of the node. To select the best split, the formula for variance is applied

$$Variance = \frac{\sum(y - \mu)}{n} \tag{3.4}$$

where μ is the mean value of the sample group. The higher the variance value, the greater the purity of the node.

3.1.4. Pruning

When building decision trees, attention should be paid to their size. In order to prevent trees from being too big, i.e. having too many nodes and leaves, a method called pruning is used. In addition, preventing the tree from growing too large, has a direct impact on its accuracy. In such case, it is very likely that overfitting will occur, i.e. the tree will be learned so well on the training data set, that it will return a wrong response on unseen new data. It is therefore important, that the tree generalizes the data well enough, to return appropriate predictions even for completely new data. For this purpose, a method called pruning is applied, which removes redundant subtree and useless split, and replace it with a leaf node. In general, pruning itself, divides into two types: pre- and post-.

Another word for pre-pruning is the early stopping rule. It is applied at the subtree construction stage, affecting its parameters such as the maximum depth or number of leafs in particular node. Another possibility is to determine the cessation of tree growth, based on the measures mentioned above. If the result does not improve, or the desired value is reached, the tree stops growing. With this approach, however, there is a risk that the tree will stop growing prematurely.

In contrary, post-pruning, allows the tree to fully grow without any restrictions. Just after the whole process, the number of branches and nodes are decreased, which prevents model from overfitting. The process starts with the bottom decision nodes, and based on accuracy measures decides if it might be replaced with a leaf node or be completely removed.

3.1.5. Model Ensembles

In the previous chapter, the concept of a single decision tree, or in other words a single predictive model, was introduced. For not very complex data, this approach can work well or very well. Nevertheless, there are situations where it is insufficient and it is important to increase the accuracy of the results. Therefore, it is possible to combine individual models into groups. Such connection is called model ensembles. The predictions of the single models are then aggregating, and results in final prediction. The main assumption is that each separate model performs a separate prediction, independently of the others. With many unrelated models, an bucket might be highly accurate. In principle, there are two characteristics of team building. The first creates models based on the same data,

but in a modified version or combination. The second, on the other hand, makes a final prediction based on the predictions of the set of models included in the ensemble. The accuracy of the models is determined by appropriate metrics, depending on whether the problem being solved is classification or regression.

Bootsrtap Aggregating

A bootsrtap aggregating (also known as bagging), introduced by J. Breiman in 1996 [36]. The basic principle of its working is to present the final result, based on aggregating the predictions of groups of models in ensebmle. Various voting methods are used depending on the prediction objective, which could be either classification or regression. In the case of classification, measures of central tendency like mean or median come into play. When using bagging, each tree is trained on random sample. Random sample is some smaller part of data from the whole data set. In case of this method, each smaller group is called bootstrap sample, an approximation for the actual data. The individual samples must accurately represent the inherent complexity present in the actual dataset. Each spread is used then for building separate decision trees in the random forests. As the final result the mean value of predictions of all bootstraps is taken in case of regression problem, and in case of classification problem the final prediction is made based on voting. A scheme of bagging is presented at Figure 3.2.

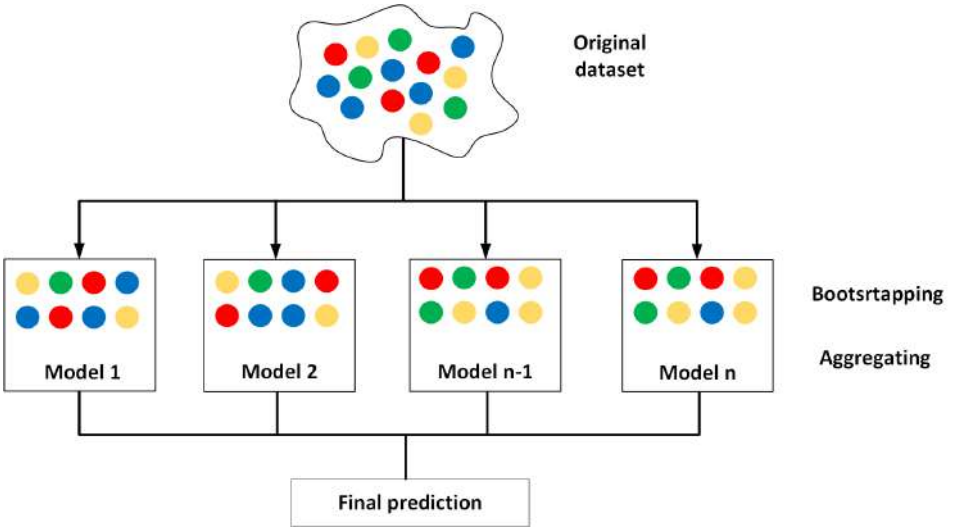

Fig. 3.2. Bootstrap aggregation scheme [72]

The final formula used to determine the final outcome of the model

$$f(x) = \frac{1}{n} \sum_{i=1}^{n} f_n(x) \tag{3.5}$$

where n stands for the number of models (also known as weak learners), and the expression $f_n(x)$ is the result from each weak learner. In here, it is obligatory to explain what a weak learner actually is. In different words, weak learners are machine learning models (Random Forests, Decision Trees, etc.) but they are good on predicting only the trained data. They do not generalize to predict accurately for all required cases. Each weak learner is built based on some sample of data. The set of weak learners, leads to create the strong learner. Strong learner is a model, that is a result of set of weak learners but with the difference, that it generalizes well to a data, and can make a accurate predictions for the whole set. The schematic working principle of the weak learners and the strong learner, which is the effect of the weaker ones, is presented in Figure 3.3.

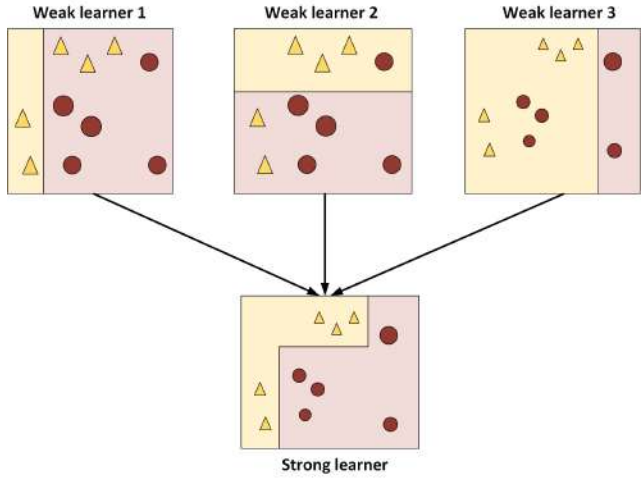

Fig. 3.3. Weak learners and strong learner visualization scheme [26]

Boosting

Boosting works similarly to bagging, on the principle of creating weak learners in order to obtain one final strong learner. The difference is in the construction, following models sequentially. Starting with building the first model, using all the data. The next step is to build a model that will be an improvement of the previous one, that is, using information about wrongly predicted samples, to improve overall performance of ensembles. The values of the weights are modified. In case of correct predictions they are decreased, and in case

of incorrect ones increased, so the more attention is paid to improve the predictions of previously missed ones. In such way the next model will predict the previously wrong samples - correctly. Each weak learner contributes to the final prediction based on its individual performance and its associated weight. The overall scheme for the construction of successive weak learners, is shown in Figure 3.4.

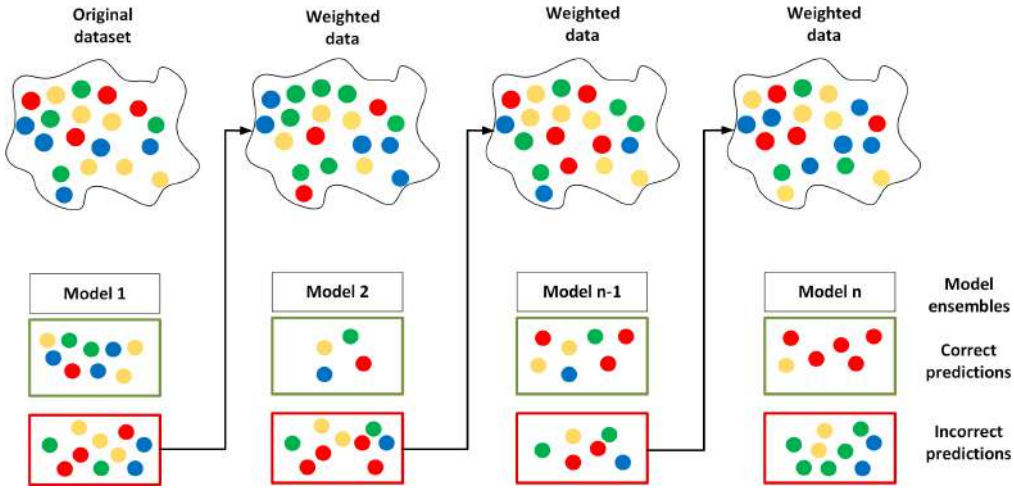

Fig. 3.4. Boosting scheme [72]

Another difference, in comparison to bagging, is the final prediction determination. While building multiple weak learners, the model predictions are assigned specific weights, and the ultimate prediction is obtained by taking their weighted average. According to the formula 3.6, the bagging method relies on a simple average of all model predictions without any weight. Therefore, the equation to determine the prediction result, using boosting method

$$f(x) = \frac{\sum_{i=1}^{n} w_i f_i(x)}{\sum w_i} \tag{3.6}$$

where w_i is the weight of the i-th weak learner. The weights are updated during the boosting process, to put mode emphasis to the more accurate models.

3.2. Multilayer Perceptron

The foundation of any neural network is a single neuron, also known as a perceptron. Before introducing deep networks, and the process of teaching them, it is necessary to discuss the principle of operation of a single perceptron. For such purpose, the simpliest

version is introduced, to provide solid background. The simplest neural network, is the so-called perceptron. Its diagram is shown in Fig.3.5. It consists of one input layer and one output node. The input layer, is formed by a matrix $\mathbf{X}$=$\{x_1, x_2, ..., x_m\}$, where m is the number of features, for a given case. For such simple architecture, a output is calculated directly from the sum of the product of weights and corresponding inputs plus some bias. Weights determines the influence of input feature to the output. Bias, is a constant, always equal to 1. They are not influenced by the results of previous layer, but they have their own output weights. This parameter guarantees, that even if all input values are equal to 0, a signal will be transmitted to activate the neuron.

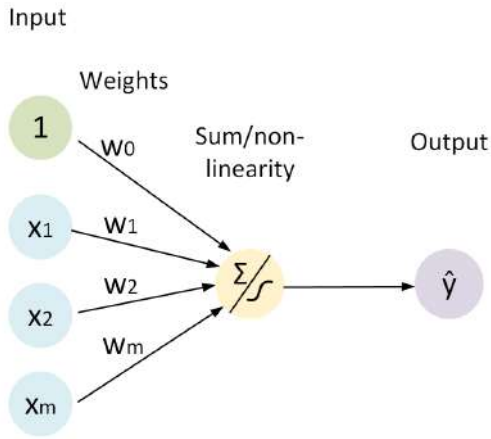

Fig. 3.5. Single layer perceptron scheme [17]

The above diagram illustrating the operation of a single neuron is described by the equation [17]

$$\hat{y} = g(w_0 + \sum_{j=1}^{m} x_j w_j) \tag{3.7}$$

where $\hat{y}$ is the output, g corresponds to activation function, w_0 is the bias, x_i and w_i are the inputs and their weights. In neural networks, a very important element is the activation function, through which non-linearity is added. It allows, to solve much more complex problems. Without this function, a neural network would work as a simple linear regression. And with it, neural networks are capable of finding nonlinear relationships between data. The activation functions used in model development are presented below. The choice of a particular function depends on the problem being solved and affects the results obtained. Graphs of some of the activation functions, namely sigmoid, tanh, RELu are presented below on Figure 3.6. The sigmoid activation function is

very popular, since its output lays between 0 and 1, and such feature is specially required in probabilistic problems where it lays between 0 and 1, as well. The ReLU activation function, characterized by picewise linearity, is especially efficient to calculate, both basic version and derivative.

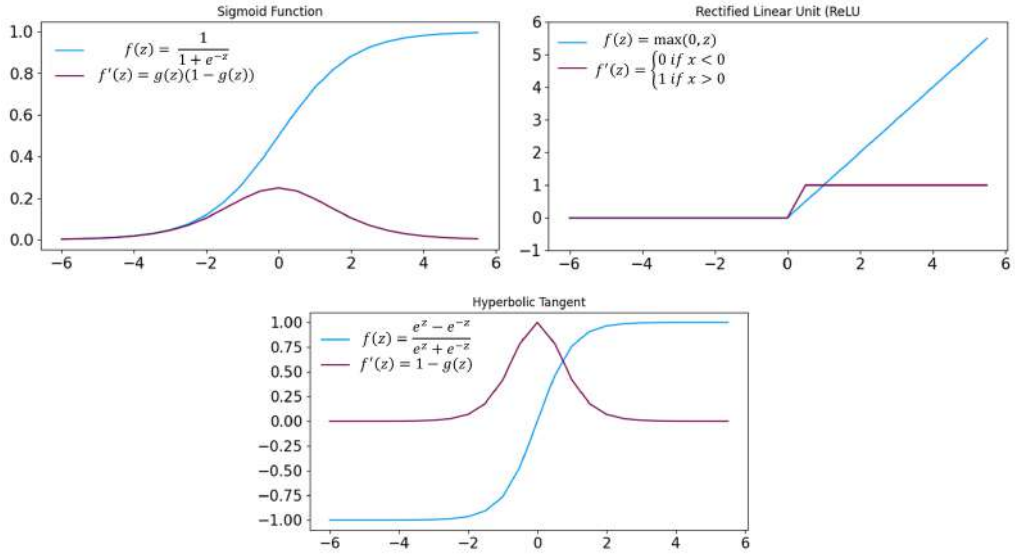

Fig. 3.6. Examples of activation functions: RELu, tanh, sigmoid [73]

Going one step further and connecting neurons together, it is possible to create a multi-output perceptron. The principle of operation is identical to that of a single one. Each output is connected directly to each input. The values of the outputs, are calculated as the sum of multiplied weights and output values, and the total is the result of multiplication with the activation function. For simplifications, the scheme was modified. The inputs remained unchanged, but as the output node a dot product of weights and input appears, and as the output of a node, we have an activation function in respect to dot product. Introducing more outputs, provides with formulas for dot product, which multiplied by activation function determines the required result

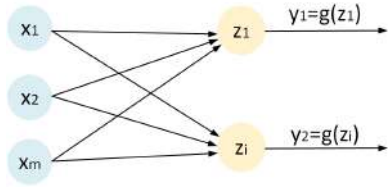

Fig. 3.7. The scheme of multioutput perceptron [17]

$$z_i = w_{0,i} + \sum_{j=1}^{m} x_j w_{j,i} \tag{3.8}$$

Individual neurons, can be combined with each other and used to predict not just one value, but several. The operation pattern of a multilayer perceptron will, in principle, not differ from that of a single perceptron. Only its complexity will be increased. Each input value will have a potential impact on the output. For the purpose of visualization, and thus better understanding of the derived formulas, the above illustration was modified.

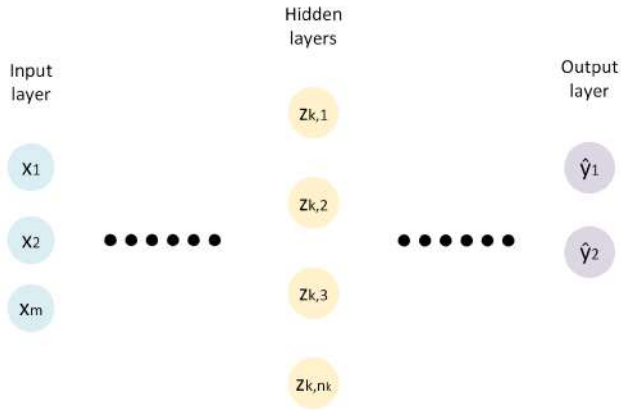

Fig. 3.8. Deep Neural Network general scheme [17]

$$z_{k,i} = {w_{0,i}}^k + \sum_{j=1}^{k-1} g(z_{k-1,j}) {w_{j,i}}^k \tag{3.9}$$

where k is the layer number. Desired outputs, are the dot products of the last layer of the neural network multiplied by an activation function according to the fomula 3.7.

3.2.1. Gradient Descent

Gradient descent is an optimization algorithm used for training machine learning and neural networks. As the goal of all ML or ANN models, such optimization, influences directly final accuracy of the model. It's main objective is to iteratively adjust the weights and biases to decrease the expected error. Before digging into GD, there is necessary to introduce an important terms like learning rate. In simple words are the steps made to reach the minimum of the cost function. The cost function, also known as a loss function, is a difference between prediction and the true value. It determined based on training data. There are two types of them:

- local minimum - the minimum parameter values within a specified range of the cost function
- global minimum - the smallest parameter value within the entire cost function

For visualization purposes, Figure 3.9A, presents the influence of learning rate step on the seeking the function minimum. The visualization presents small and big learning rate values, so two extremes. The idea is to find the most optimal one, to perfectly fit the functions minimum. In case of too small value, the calculation process might not be efficient, since more iterations have to be done. Also, the required point might not be reached before the end of calculations. However, with the bigger learning rate, there is increased chance to overshoot the desired minimum of the cost function. It is also important to adjust the learning rate, so it will not stuck in local minimum and will be able to get as close as possible to the global minimum. Figure 3.9B presents both local and global minima and maxima. It's worth to mention here, that when referring to local scenarios, it pertains to the segment of the function within the specified interval.

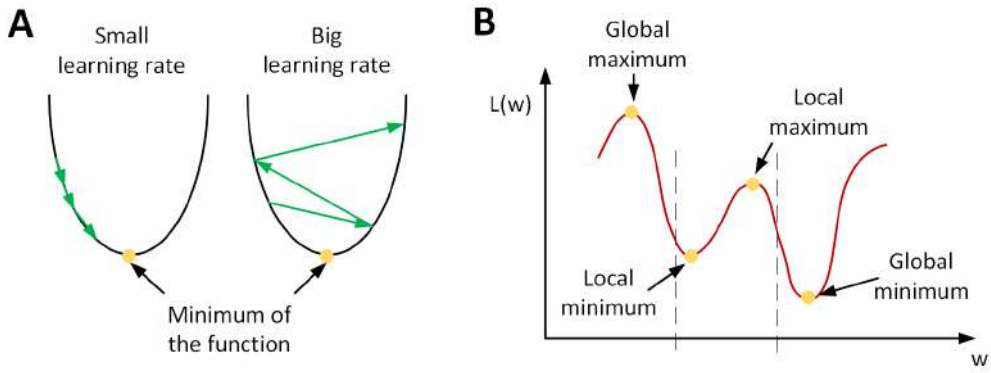

Fig. 3.9. **A**: Visualization of small and big learning rate. **B**: Visualization of a cost function, with marked global minimum and maximum, as well as local ones in specified range

Finally, a general formula, determining the change of the gradient with the change of the position along the cost function

$$w \leftarrow w - \eta \frac{dJ}{dw} \tag{3.10}$$

where J defines the cost function, w is the weight, subscripts new and old denotes weights after and before the update respectively, and η is the learning rate.

3.2.2. Backpropagation

Backpropagation is an algorithm used for training of neural network. Backpropagation carries the error backward and computes the gradient for every weight. This gradient is subsequently used within the gradient descent process, which lets to update weights in the process of learning. In case of the final layer (the output layer), determining the error is relatively easy, since the final output is known. In case of hidden layers, the task is not that trivial, since the target values in each neuron in hidden layers are not directly known. In process of training neural networks, with gradient descent, the gradient of the error function in respect to weights and biases has to be calculated. Each iteration of gradient descent, changes their values. A back propagation algorithm for a multilayer neural network will be presented below. For introductory purposes, the chosen error function is the mean square error:

$$E(X, w) = \frac{1}{2N} \sum_{i=1}^{N} (\hat{y}_i - y_i)^2 \tag{3.11}$$

where y_i is the true value and $\hat{y}_i$ is the predicted value.

A backpropagation algorithm is based on chain rule, and product rule from differential calculus. The effectiveness of applying these rules heavily relies on the choice of activation functions. It it crucial for the selected activation functions to be differentiable. In the deriving of equations, a bias will be taken into account, and as in previous sections it will be denotes as w_{0i}^k with fixed output $o_0^{k-1} = 1$ in layer $k-1$. According to such assumption

$$w_{0i}^k = b_i^k \tag{3.12}$$

Going forward, we reproduce equation 3.9 in form

$$z_i^k = b_i^k + \sum_{j}^{n_k} w_{ji}^k o_j^k - 1 \tag{3.13}$$

The main goal, for using backpropagation is to minimize the cost function error with respect to the network weights, by calculating the derivative. The error function can be separated into sum over individual error terms for each pairs of inputs corresponding to outputs. The derivative is calculated with respect to each pair. At the end, the sum of derivatives of each functions is determined

$$\frac{\partial E(X, w)}{\partial w_{ij}^k} = \frac{1}{N} \sum_{i=1}^{N} \frac{\partial}{\partial w_{ij}^k} \left(\frac{1}{2} (\hat{y}_i - y_i)^2 \right) = \frac{1}{N} \sum_{i=1}^{N} \frac{\partial E}{\partial w_{ji}^k} \tag{3.14}$$

The derivation of the backpropagation algorithm starts with applying the chain rule to the error function partial derivative

$$\frac{\partial E}{\partial w_{ji}^k} = \frac{\partial E}{\partial z_j^k} \frac{\partial z_j^k}{\partial w_{ji}^k} \tag{3.15}$$

where z_j^k is the formula, before passing it through activation function. The first term, of the above equation can be also denoted as

$$\delta_j^k \equiv \frac{\partial E}{\partial z_j^k} \tag{3.16}$$

The second term from the equation 3.14 is determined as

$$\frac{\partial z_j^k}{\partial w_{ji}^k} = \frac{\partial}{\partial w_{ij}^k} \left(\sum_{l=0}^{n_{k-1}} w_{lj}^k o_l^{k-1} \right) = o_i^{k-1} \tag{3.17}$$

According to the above, the derivative of the error function to respect to a weight is

$$\frac{\partial E}{\partial} w_{ji}^k = \delta_j^k o_i^{k\text{‘}} \tag{3.18}$$

The partial derivative is the of a weight is a product of the error term $delta_j^k$ at node j in layer k and the output o_i^{k-1}.

Moving towards the output layer, applying the partial derivative, and applying the chain rule

$$\delta_1^m = (g_0(z_1^m) - y) g_0^{'}(z_1^m) = (\hat{y} - y) g_0^{'}(z_1^m) \tag{3.19}$$

As the final, the partial derivative of the error function E withh respect to a weight in the final layer w_{i1}^m is determined as

$$\frac{\partial E}{\partial w_{i1}^m} = \delta_1^m o_i^{m-1} = (\hat{y} - y) g_0^{'}(z_1^m) o_i^{m-1} \tag{3.20}$$

For hidden layers, the partial derivative of the error function E with respect to a weight in the final layer w_{ij}^k, with an assumption that $1 \leq k < m$ is formulated as

$$\frac{\partial E}{\partial w_{ij}^k} = \delta_j^k o_i^{k-1} = g^{'}(z_j^k) o_i^{k-1} \sum_{l=1}^{n_{k+1}} w_{jl}^{k+1} \delta_l^{k+1} \tag{3.21}$$

To conclude the backpropagation involves computing the gradients of the cost function concerning the weights. It is employed to enhance the outcomes of neural networks. It is achieved by propagating the error in a reverse direction, determining the gradient of the cost function for each weight. These gradients are then used in the gradient descent process to refine the neural network's performance.

3.3. Metrics

One of the most important methods to determine whether a built model is reasonably accurate, is to select a specific metric. This is a key element, which is most often, one of the criteria for considering that a given prediction is accurate enough. The choice of the appropriate metric, is dependent on the problem being solved, and should always be selected individually. Nevertheless, it happens that several variants are checked to make sure that the results obtained are reliable. The choice of metrics also depends on the type of problem being solved. More specifically, whether it is a regression or classification problem. In this work, problems in the first mentioned group are solved, so the metrics corresponding to it, will receive more attention.

3.3.1. Mean Absolute Error

The mean absolute error is the average of the difference between true values and their predictions, defined as [106]

$$MAE = \frac{\sum_{i=1}^{n} |y_i - \hat{y}_i|}{n} \tag{3.22}$$

where y is the true value, $\hat{y}$ is the prediction, n determines the number of values. MAE metrics is easy to be understood, as it describes the mean value of the error (under or over estimation) of the prediction.

3.3.2. Mean Squared Error

Mean squared error [71] is the average squared distance between the observed and predicted values. Because of the square, the interpretation of the metric is less intuitive. Also, it pelizes the single bigger over predictions. The MSE formula is determined by

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{3.23}$$

Another interpretation of it, is the root of the resulting value referred to as RMSE (Root Mean Squared Error)

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{3.24}$$

3.3.3. Coefficient of determination

In general the coefficient of determination, is in statistics, the portion of variation in the dependent variable that is predictable due to independent variables. To determine its value, first the mean value of observations has to be determined

$$\bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i \tag{3.25}$$

where $\bar{y}$ denotes the mean of true values. Residual Sum of Squares (RSS) is a statistical method, that might be used to determine the level of discrepancy in a dataset not predicted by a regression model. It measures the variance in the observed data, compared to the predicted values. RSS indicates, if the regression model fits the real dataset, and how well.

$$RSS = \sum_{i=1}^{n}(y_i - \hat{y}_i) \tag{3.26}$$

The sum of squares total (SST), is the squared differences between the observed dependent variable and its mean. In simple means it is the dispersion of the observed variables around the mean. The formula is determined as follows

$$SST = \sum_{i=1}^{n}(y_i - \bar{y})^2 \tag{3.27}$$

Combining the above equations leads to the final equation for R squared, which indicates the correlation between the true and predicted values

$$R^2 = 1 - \frac{RSS}{SST} \tag{3.28}$$

4
Experimental test-rig for tracking particle collisions

4.1. Experimental test-rig

Whenever performing numerical simulations, it is crucial to compare the outcomes with real-world measurements. It ensures that the calculations accurately represent reality, or how far they deviate from it. They additionally provide the opportunity to fine-tune the models to achieve results, that closely resemble reality. To validate the novel HELSCM approach, an experimental setup was established to gather measurements, which were subsequently incorporated into the simulations. It enables a comprehensive assessment and comparison of the obtained results.

The experimental test rig used for measurements, is located at the Silesian University of Technology in the Department of Thermal Engineering laboratory. Figure 4.1A, with key components being identified and described for further clarity. It was designed and produced, based on the numerical model discussed in [104], which was used to test the effect of particle collisions, depending on the coefficients of the collision model. As these were only numerical calculations, several corrections were made to the real device. The construction was 3D-printed, which ensured a high degree of accuracy for the carefully designed gas flow paths. The transparent core, i.e. the main particle collision region, was made from the transparent polycarbonate.

The airflow was controlled by three precise Bronkhorst HIGH-TECH B.V. [4] mass flow controllers. The two smaller units (BH1 and BH2), operating in the range of $1.2-60.0$ ln/min (F-201AV-50K-VBD-00-V), were used to control the flow rate in the side injection tubes (namely left and right tubes), marked at Fig 4.1B. The air flow in the main tube, was controlled by the largest unit (BH3), operating in the range of $7.2-360.0$ ln/min (F-203AV-M50-VBD-00-V) (Fig. 4.1B). The flow rate is determined using the output unit on the controller, expressed in liters per cubic meter ($\frac{l}{\mathrm{m}^3}$) under standard conditions (ln/min determined at $T_n = 273.15$ K and $p_n = 101325$ Pa).

To record the measurement, a high-speed camera was situated in front of the test-rig. It was set up to record the particle flow in the core, where two streams of particles collide. The high speed camera VEO 710L [8] was equipped with a NIKKOR 200 mm F/4.0 MACRO lens. Particles were tracked with the resolution of 1280x720 and the frame rate of recorded images was set to 8,300 frames per second (fps). Such camera set up, allowed to capture the collision region at one image (without necessity to divide it in

two), Such solution, allowed to avoid the inaccuracies associated with the final merging of images, which was earlier problematic and discussed in paper [139]. To ensure sufficient illumination for the camera, a back-light (LED Phlox HSC Phantom 10 x 10 cm) was used with a luminance of $\geq$130,000 cd/m^2, uniformity of $\geq$ 95%, and light temperature of 5,700 K. In addition, a sample image, taken during the measurements, was superimposed on the test-rig scheme (4.1). The sections, where the time-averaged particle velocities were determined, are also marked on it. The entire procedure will be described in Section 4.2.

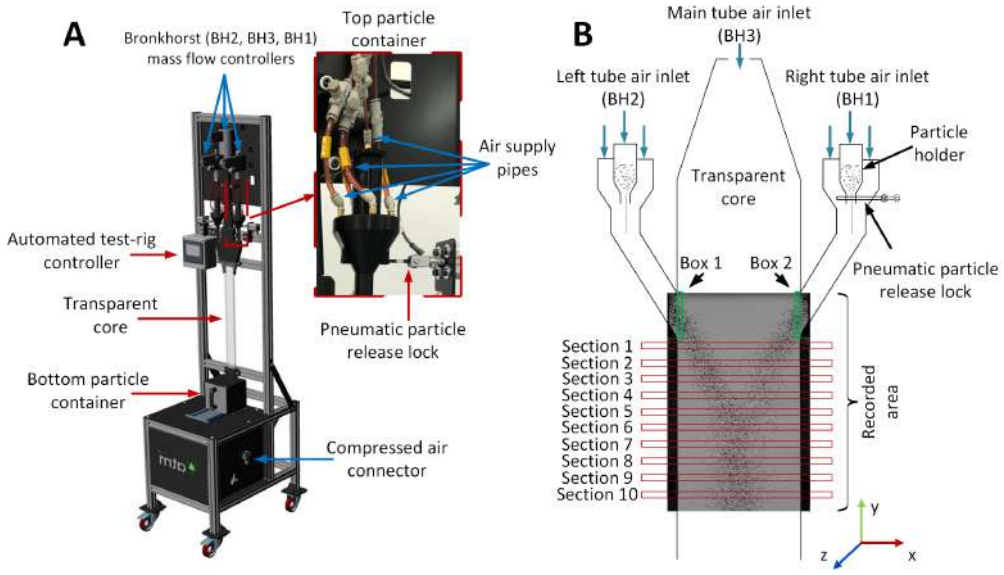

Fig. 4.1. **A**: The configuration and description of experimental test rig. **B**: Schematic overview od the test-rig, with marked key elements, and the superimposed image from the high-speed camera measurement recording

One of the key components of the test-rig are the particle containers, which are refilled before the start of each measurement. It is the particles from the containers, that are analyzed for collisions. The particle containers are ending right and left side inlets. Each one, easily stores 15 g of material, which allows for approx. 3 s of testing. The air, to the side tubes is delivered through five separate canals. One of them, is attached to the particle holder itself, to ensure particle release, by creating excess pressure. The proportion of introduced air is 10% of the total supplied to one of the sides.

The measurement procedure was automated and controlled using Siemens SIMATIC S7−1200. The whole process started with the operator filling the holders, and turning on the air flow. Next, few steps in sequence were automated:

– the core flow continued for 2−5 s, until it was stabilized,

– the side inlets flow continued for 5−10 s, until it was stabilized,
– when each flow was stabilized, the plate blocking the particles from spilling out too early was moved, thus allowing the particles to fall out of the container,
– the image recording started.

Once, the containers were completely emptied, the operator turned off the airflow. Each of these steps was performed for each of the nine measurements taken.

The selected test material was gray polyethylene microsphere particles [6], with the density equal to 1000 kg/m^3. Such type of material is characterized by an almost perfectly spherical shape. Table 4.1 contains the particle size data provided by the manufacturer. The initial conditions for the cases under investigation were set as follows: a flow rate of 38.4 ln/min for the side tubes and 240.0 ln/min for the core tube.

Table 4.1

Particle size information of the polyethylene microspheres

	300−355	425−500
Largest diameter, μm	354	520
Smallest diameter, μm	300	429
Mean diameter, μm	315	475
Median diameter, μm	314	473
Standard deviation, μm	8.2	22

Nine measurements were carried out for each diameter range, shown in the table 4.1. The measurements were carried out several times, to ensure repeatability. Despite the differences in diameter distribution, a common one was the velocity of the provided air. This unit is the one used by the Bronkhorst devices in the test setup. In the subsequent sections, these aforementioned values were converted to correspond to a gas velocity of 20 m/s, for main and side inlets.

4.2. Image analysis

Granular flows, where the number of interactions between particles is really high, are very demanding in terms of their tracking. For this purpose, a dedicated application was created using the National Instruments LabVIEW software. In this case, the application, allows the particle velocity to be analyzed efficiently and accurately, so it can be compared with the results of numerical simulations in a subsequent step. The particle velocities are determined within 10 frames (whose positions are well defined, and coincide both in the experiment and the numerical calculations). The development of a program determining the particle velocity was necessary, to analyze the large number of images from the measurements. and to determine the average speed of the particles, in the sections of interest. Additionally, the program allows to follow the same particle in successive steps.

The program converts grayscale image into a binary image with a defined threshold, which can slightly differ for simulated and experimental data. The threshold, defined at the beginning of the conversion procedure, remains constant for all analyzed cases. To check the influence of the used threshold range, two configurations of its valued were investigated. In the first step, two values (minimum and maximum) were chosen, when the particles were seen without iterating with the background. The second step, was to take the threshold value, from the middle of the set range. Checking for the effect of used threshold, on changing camera position or backlight system, showed that the threshold value range was negligible. However, it should be borne in mind, that every time the settings or positions of the equipment are changed, it is necessary to carry out a check. For the measurements carried out, in the current work, the settings and location of the equipment were not changed. The position of detected particles were determined using binary image. Searching for particles in an image is expensive, so in order to reduce computation time and cost, the area was split so the search was restricted to a specific frame. These, detected during one of the analyses, in the relevant sections and smaller boxes, are highlighted in green in Figure 4.2. To ensure the clarity of the visualization, only three sections where the average velocities were highlighted on the picture. For further calculations, and result presentations ten were used.

The working principle of the program is illustrated in Figure 4.3. After the position of the particle at a given time step is determined, the tracking procedure is activated. The process begins with the determination of maximum and minimum distance, in length ($x_{\max}$) and height ($y_{\max}$), that the analyzed particle can shift. Determining those two values, consisted of manually observing the moving particles in successive images, in defined frames. The vertical edge of first tracking box, a blue box, as marked on Figure 4.3), starts in the middle of a particle.

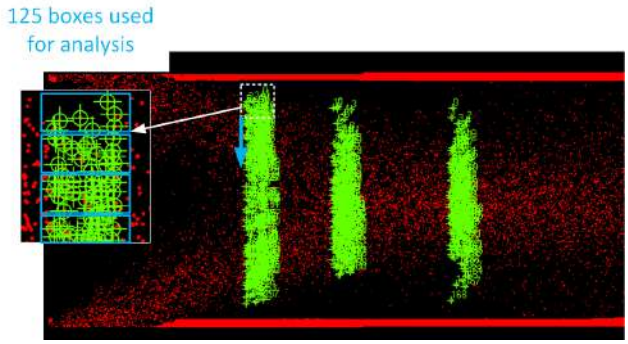

Fig. 4.2. Particles detected from the binary image highlighted in green with the analysis boxes used for particle tracking

In the next step, in next tracking box (a red box) again the same particle is detected. Determining, and being sure, whether it is exactly the same particle being analyzed can be debated. There is no one hundred per cent certainty that in a particular box, there is not a particle that travelled a much greater distance than assumed, or whether a particular particle is exactly the one it was assumed. As a precaution against such a situation, a maximum value was defined for the vertical and horizontal distance a particle can travel. This is an issue that can be discussed and questioned. There are limitations, which were made on the basis of manual observations of the particles. However, without these, it would be impossible to analyze, several thousand consecutive, images to determine the average velocity of the particles concerned. Another possible complication is, when more than one particle appears in the box under investigation. It might increase the likelihood of particles being mistaken for each other. The applied solution is, to count the number of particles in each frame. If their number increased of more than 10%, the size of the investigated frame is adjusted. In addition, the minimum path taken by the particles was determined, during manual analysis, of several images. If one of the particles travels less than minimal distance, it is eliminated from the analysis.

To determine the velocity profiles, each of the ten sections, was divided into a certain number of boxes (see Fig. 4.2) at its height. The described tracking procedure, determines the average particle velocity in particular box, and stores it in memory. Despite conducting calculations in stabilized flow, particle velocities fluctuate over time a bit. To minimize this effect, the time-averaged (TA) was used, to determine the time-averaged velocity in particular boxes. Such velocity was determined as

$$\phi_{time,i+1} = \phi_{\mathrm{time},i} + \phi_i \cdot \Delta\tau_i u_{TA} = \frac{\phi_{time,i} - \phi_{time,i-n}}{\tau_i - \tau_{i-n}} \tag{4.1}$$

where ϕ denotes the averaging variable — in this case, it is velocity — and u_{TA} denotes the time-averaged velocity. These equations allow the time-averaging of any variable during simulation or post-processing. To verify whether the flow has stabilized, the average particle velocity fluctuations were monitored over the given time steps.

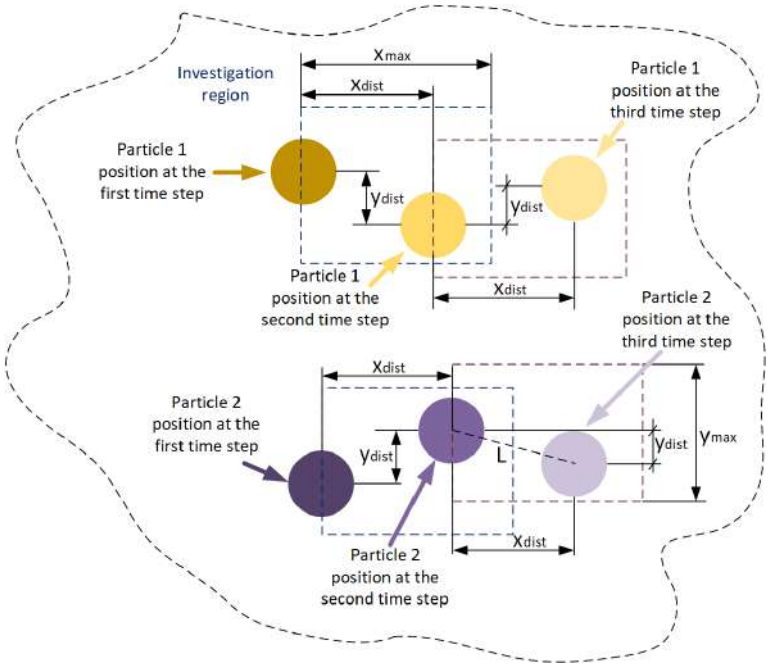

Fig. 4.3. The procedure for determining the position of a given particle and its displacement to determine the average velocity

4.3. Uncertainty analysis of the experimental data

When carrying out measurements, it is very important, to determine the uncertainty, resulting from the uncertainty of the equipment used. It is also important, when such results are compared with the results of numerical simulations. Without such data, it is hard to fit numerical model into exactly the same results, as those coming from an experiment. With a specified uncertainty, it is possible to determine, the error in the measurement results. Such uncertainty is applied to the measurement points as error bars. The superimposed results, from the numerical calculations, should be within the specified range.

One of the devices, causing some uncertainty is the high-speed digital camera. For the used device, the factors affecting uncertainty are pixel size (u_x = 0.02 mm) and image acquisition rate, calculated as

$$u_\tau = \frac{\Delta u_{TA}}{\Delta \tau_{TA}} \cdot \Delta \tau \tag{4.2}$$

where Δu_{TA} is the maximal velocity change, during the period of time-averaging (TA) calculated for $\Delta \tau_{TA}$ equal to 0.0321 s, corresponding to 450 frames, $\Delta \tau = 7.14286E-05$ s is the time step between subsequent frames. The uncertainties due to the measured velocity depend on the accuracy of the mass flow controllers (FM) u_{FM}, equal to $\sigma_{FM} \pm 0.5\%$. The uncertainty u_{FM} can be calculated as

$$\begin{aligned} u_{u,min} &= \frac{m_{min}}{A_{tube} \cdot \rho_{air}} \quad \text{where} \quad m_{min} = \frac{V_{n,min} p M_{air}}{RT} \\ u_{u,max} &= \frac{m_{max}}{A_{tube} \cdot \rho_{air}} \quad \text{where} \quad m_{max} = \frac{V_{n,max} p M_{air}}{RT} \\ u_{FM} &= u_{u,max} - u_{u,min} \end{aligned} \tag{4.3}$$

where p is the pressure equal to 101.325 kPa; T is the temperature of gases within the rig ∼298 K, M_{air} is the atomic weight of air 28.84 kg/kmol, R is the universal gas constant (8314 J/(kmol·K)), ρ_{air} is the air density under real conditions, and $V_{n,min}$ and $V_{n,max}$ are the volumetric flow rates under normal conditions (p_n = 101325 Pa, T_n = 273.15 K). The methodology was applied to the mass flow controllers attached to the tube injectors $u_{FM,tube}$ and core section $u_{FM,core}$. The total uncertainty of the measured gas velocity u_u is the result of the law of error propagation, which is defined as [81]

$$\begin{aligned} u_u^2 = &\left(\frac{\partial u}{\partial x}\right)^2 u_x^2 + \left(\frac{\partial u}{\partial \tau}\right)^2 u_\tau^2 + \left(\frac{\partial u}{\partial l}\right)^2 u_x^2 + \\ &2\frac{\partial u}{\partial x}\frac{\partial u}{\partial \tau}\mathrm{cov}(x,\tau) + 2\frac{\partial u}{\partial x}\frac{\partial u}{\partial l}\mathrm{cov}(x,l) + \\ &2\frac{\partial u}{\partial l}\frac{\partial u}{\partial \tau}\mathrm{cov}(l,\tau) + u_{FM,tube}^2 + u_{FM,core}^2 \end{aligned} \tag{4.4}$$

where subscript x, τ, and l refer to pixel location, time, and length (in the x direction: x_{dist}) of the box used for the velocity calculation. The covariance cov maintains the correlation between time and space. For the considered case, those quantities are not correlated, so the covariance disappears and the equation simplifies to

$$u_u = \sqrt{\left(\frac{\partial u}{\partial x}\right)^2 u_x^2 + \left(\frac{\partial u}{\partial \tau}\right)^2 u_\tau^2 + \left(\frac{\partial u}{\partial l}\right)^2 u_x^2} + u_{FM,tube} + u_{FM,core} \tag{4.5}$$

For two analyzed cases, the final error value for measurements of colliding particles with mean diameter equal to 315 μm was 0.808 $\frac{m}{s}$, and for mean particle diameter equal to 475 μm error was 0.756 $\frac{m}{s}$. The Table 4.2 presents the separate errors, which, when combined, yield the final measurement uncertainty for both scenarios, i.e. for two different particle sizes.

Table 4.2

Calculated uncertainties for investigated particles group, based on the law of error propagation

Mean particle diameter, μm	$\frac{\partial u}{\partial x}$, $\frac{1}{\mathrm{s}}$	$\frac{\partial u}{\partial \tau}$, $\frac{\mathrm{m}}{\mathrm{s}^2}$	$\frac{\partial u}{\partial l}$, $\frac{1}{\mathrm{s}}$	u_τ, s	$u_{FM,tube}$, $\frac{\mathrm{m}}{\mathrm{s}}$	$u_{FM,core}$, $\frac{\mathrm{m}}{\mathrm{s}}$	u_u, $\frac{\mathrm{m}}{\mathrm{s}}$
315	4.679	13.067	6598.8	9.33E-04	0.199	0.434	0.808
475	3.565	9.955	4907.2	7.11E-04	0.199	0.434	0.756

5
Development of simplified collision approach

Models such as the HEL or DEM have their advantages and disadvantages. In the case of the former, the precision depends on the determined value of granular pressure. Its determination often causes instabilities during simulation. Despite, above mentioned point, an undeniable asset is the calculation time and the ability to simulate flows where the particle diameter varies. It is an option used on a daily basis, to simulate flows in large industrial facilities. The second approach considered, is the DEM model, which, despite its accuracy in collision detection, is not widely used. This solution is time-consuming and computationally demanding, making it an option only for simulating flows with small numbers of particles, in geometries of small size. Knowing the capabilities and limitations of both models, as well as observing the evolving world of machine learning and artificial intelligence [62, 63, 74], the idea of using ML techniques to combine the advantages of the approaches, and create an alternative approach was born. The main idea is to create a simplified model (SM), using ML algorithms, and then apply the model to an existing HEL approach, where the term F_{coll} in equation (2.26) will be replaced. With this combination, the problem of calculation stability, caused by the need to determine the granular temperature will be eliminated. In addition, thanks to the speed of prediction of new collision forces, the time of the simulations themselves will be significantly reduced. The new approach is called Hybrid Euler-Lagrange Surrogate Collision Model (HELSCM).

5.1. Data set generation procedure

The key linking the two CFD models, i.e. HEL and DEM, is the machine learning algorithm. Before building it itself, then important step is to collect a large amount of data so specific inputs correspond to specific outputs. The validated DEM model [140] was used for this purpose. The process of collecting the data is visualized in Figure 5.1. The simulations took place in a domain, where only two particles collided at a time. At the start of the simulation, particle information, the velocity vector components ($\mathbf{u}$, $\mathbf{v}$, $\mathbf{w}$) were saved to a file. After a certain number of time steps, the corresponding values of the velocity vector components, were saved after the collision. In the presented solution, additionally the model stores information about particles, i.e. diameter, density. To not introduce too many complexities in single step expected of velocity components remaining

data is only stored for future application. The data generation process is case-specific. Therefore, it is built for specific considered conditions and cases, i.e. in this situation, the model is built for combinations, as close as possible to those, occurring during carried out experiment. A low-scale DEM model was used for data generation, to model particle interactions, implemented in ANSYS®Fluent, the Cundall and Stack [48] approach was used. Choosing the least complex approach, when carrying out simulations (i.e. small domain, only two particles analyzed), reduces the likelihood of error, and gives much more control over the process.

Before the data generation, the velocity range (V_{mag}), at which the particles can move were determined. With regard to the experiment carried out and the results obtained, the minimum value is 0 m/s, and the maximum is equal to 20 m/s (velocity calculated from gaseous phase). The set of equations used for the calculation of the velocity components vector for the defined angles φ and θ based on the velocity magnitude of the particle is defined as

$$\begin{aligned} u_{A,B} &= V_{\text{mag}} \cdot \sin\theta \cdot \cos\varphi \\ v_{A,B} &= V_{\text{mag}} \cdot \sin\theta \cdot \sin\varphi \\ w_{A,B} &= V_{\text{mag}} \cdot \cos\theta \end{aligned} \tag{5.1}$$

where φ is the angle defined from the OX axis, and θ is the angle defined from the OZ axis, as illustrated in Figure 5.2A. Figure 5.2B presents the distribution of particle velocity configurations, after determining the components of the particle velocity vectors. Each particle directs to the origin of the coordinate system. With this assumption, it is certain that the particles are moving in the same direction, and will therefore collide. For clarity of the diagram, the velocity vectors and the directions of the particles were not marked.

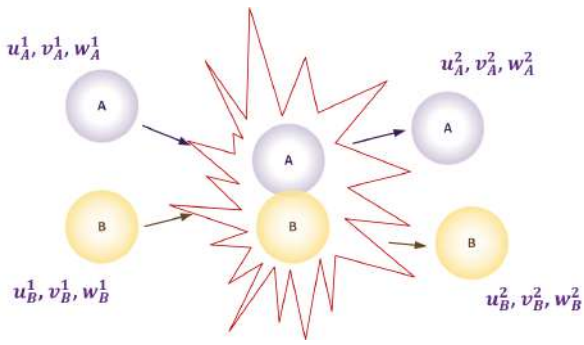

Fig. 5.1. Configuration of the low-level collision model used for collecting data set for ML model

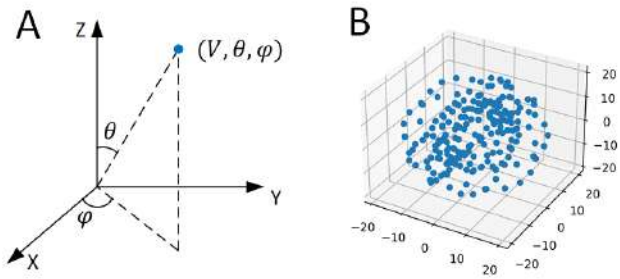

Fig. 5.2. **A:** Visualization of the methodology used for the calculation of the velocity components based on spatial particle location and its velocity magnitude, **B:** distribution of the points that corresponds to particle position according the velocity vector configurations pointed into the center of coordinate system

Based on the components of the velocity vectors, the positions of the particles were determined. Recalling, the important assumption is the particles in each simulation, at the beginning of the simulation, headed towards the centre of the coordinate system. In this way, only the cases where a collision occurred were fed into the model, without introducing additional, unnecessary data that could negatively affect the accuracy of the model. In addition, the overall simulation time is significantly reduced. An additional factor that may be undesirable, is an unnaturally large overlap of particles. When using a DEM model, such a situation results in an unnaturally large deformation, which exaggerates the velocity value resulting from the collision. Therefore, a default limitation of the post-collision velocity was set, equal to 100 m/s.

The most important step, was to identify and generate collision cases so that possible configurations would appear in the database. For visualization, a Figure 5.3 was created, where possible collision configurations of two particles is presented. Three key moments are presented, where the first is the initial position of the particles with velocity vectors marked, the moment of collision, and the position of the particles after the collision, also with velocity vectors marked. The particles pre- and post-collision position, as well as velocity vectors values might be exaggerated. One possible case of collisions is when the particles have the same turn, but differ or not in their velocities, as presented in Figure 5.3A). Another possibility is parallel collisions, where particles move at different speeds and have exactly the same returns. These are cases, where one particle hits the other from behind, transferring some of its energy (refer to Figure 5.3B). There are no cases where the particle behind has a lower velocity than the one in front of it. In such a situation, the collision would not take place. Another configuration is the case when two particles have opposite directions, these are so-called head-on collisions (Figure 5.3C).

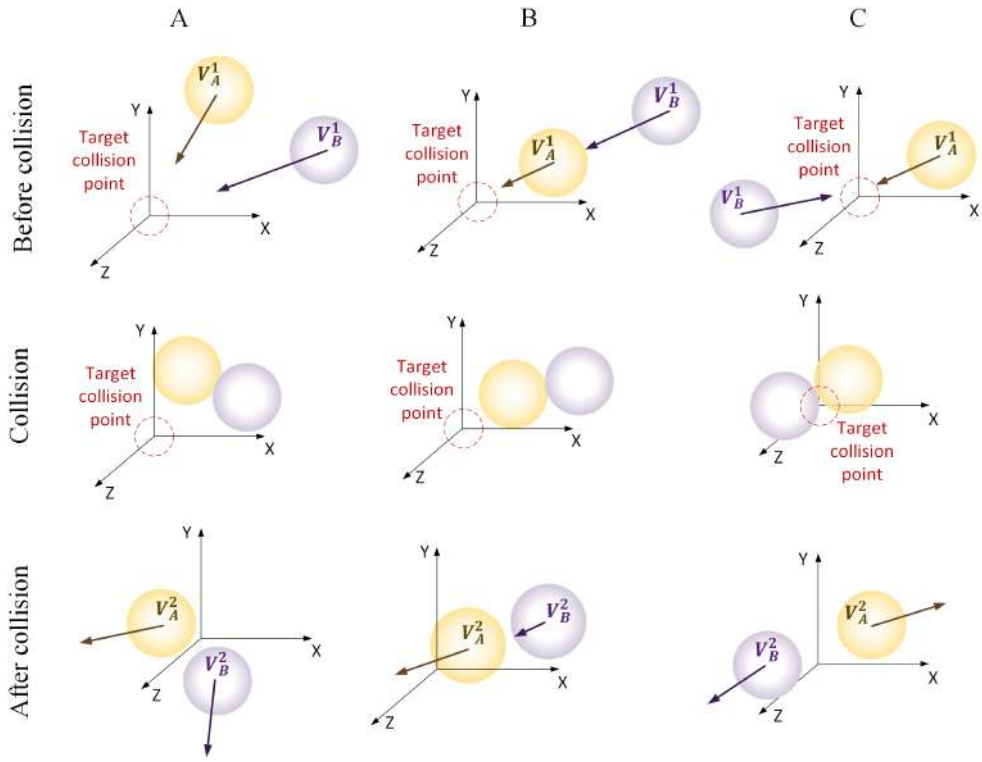

Fig. 5.3. Approximate visualization of possible collision configurations and the stage before, at and after the collision. Column A shows a possible collision of particles moving in the same direction, column B of particles moving in so-called parallel and column C of head-on collisions

Due to the very large number of possible cases (approx. 60,000), the process of data generation, was fully automated with a group of scripts, written in Python. The simulations themselves, were performed in Ansys® Fluent software. Following the flow chart 5.4, the process started with reading a file, with saved all possible collision configuration. The information contained therein, is required to be given in exactly the same form to the injection file acceptable by Ansys® Fluent. Then 100 collision cases were selected, from which 200 injection files were created (two files, each corresponding to one particle, were needed for one simulation). Once specific cases were selected, the program created a journal file so that automatically when one simulation ended, another one start with new initial conditions for subsequent particles. Such a step was necessary to reduce the time needed to generate all the data. It would be too time-consuming to run the Ansys® Fluent program from scratch each time, given such a large number of calculations to be performed. Therefore, the decision was made to run the program from scratch only once, for every 100 simulations. In addition, such a solution, provided a safeguard in

case of unplanned failures in the operation of the program (caused by random situations, such as power supply problems, etc.). Another important element, was to create a User Defined Function (UDF), which at the end of each calculation, at the appropriate time step (guaranteeing the moment after the collision) wrote post-collision information to a file. This function also took care of saving the appropriate number of the calculated case. With such a large amount of data, a huge emphasis had to be put and it was necessary to make sure that the output would definitely correspond to specific inputs. Any mistake at this stage, would generate false data, and thus inaccuracies would be introduced into the future model.

Another of the speed-ups used, was to split the main configuration file into 10 smaller ones. It allowed a separate Fluent to be run for each subgroup, making the calculations for each subgroup run in parallel. Such solution, roughly speaking, reduced simulation time by about 10 times. The division did not affect changes in the order of activities performed, for each subgroup the process looked identical as presented in the flow chart 5.4. After many attempts, the time required to generate data, for so many configurations, was reduced to about a week. This is a big improvement, in case the need would arise to generate a new set, for different data.

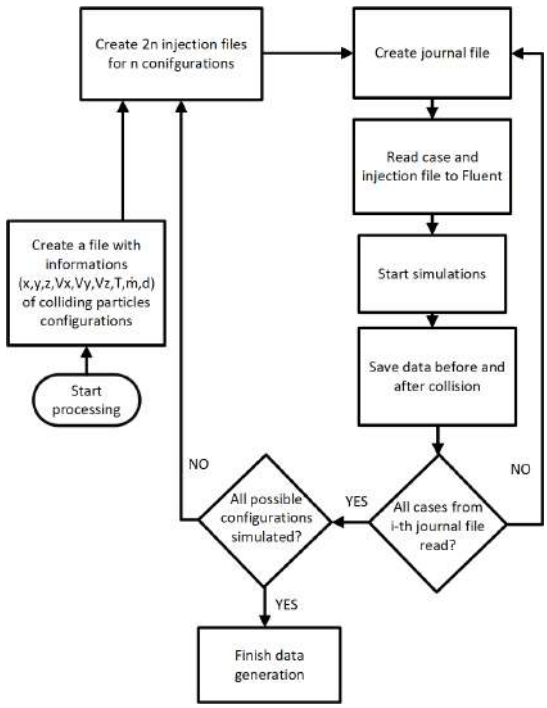

Fig. 5.4. Flow chart presenting automation of the data generation process

5.2. Development of a Surrogate Model

When building any predictive model, the entire data set at one's disposal must be divided into essentially three subsets: train, validation and test. In the process of learning and model building, only the first two of the aforementioned are involved. On the training set, the model is learned. The validation set is used to determine whether the training went satisfactorily or whether the learning process still needs to continue. In the case described, 5% of the data was allocated to the test set, 25% of the rest to the validation set, and 75% to the training set. Once the data was divided, it underwent a standardization process [89]. In the data the values vary from 0, all the way up to 20, which affects the large difference between the values that occur. Standardization is taken into account to reduce the wide distribution of values, so that the contribution of each feature entering the model is approximately proportional. To perform the standardization, the mean values and standard deviations for each characteristic, and for the outputs, must be determined according to the formula [93]. Here, it is important to remember that these values are determined on the basis of the training set, and then to carry out the standardization for training, validation and test sets.

$$\mathbf{X}_{\text{sc}} = \frac{\mathbf{X} - \mu_{\text{train}}}{\sigma_{\text{train}}} \tag{5.2}$$

where $\mathbf{X}_{\text{sc}}$ is the standardized value, $\mathbf{X}$ is the original value, μ_{train} stands for mean and σ_{train} stands for standard deviation.

Each model that is built, is characterized by certain hyperparameters, the values of which determine its accuracy of precision [44]. Hyperparameters, and their values, strictly depend on the data with which the models are built. Searching for the best hyperparameters can be done in various ways, e.g. manual space search (less optimal solution, requiring users activity), grid search (automated version of manual search, works on the same basis but the values of of hyperparameters are changed by the function), grid search (the values to be tested are specified, the whole process is automated, but still every configuration is checked), random search (similar to the grid search, with the difference that not all possible configurations are checked but random ones), or used in presented work Optuna [19]. Optuna formulates hyperparameters optimization as a process of minimizing/maximizing an objective function, taking a set of hyperparameters as input and returning with its (validation) score. The tool uses Bayesian optimization. The algorithm searches for hyperparameters from the space, where they give the best results, in case they are not satisfactory the space is not searched. With this approach, it is quicker to give up on analyzing configurations that do not give promising results. During the search for hyperparameters, pruning is also used, which is responsible for interrupting

calculations when there is no improvement in accuracy. Such an approach reduces the cost of computation, further reducing the search time.

The hyperparameters that were chosen to optimize, were below mentioned five

- n_estimator − number of gradient boosted trees, equivalent to the number of boosting rounds,
- max_depth − maximum tree depth for base learners,
- learning_rate − boosting learning rate,
- colsample_bytree − defines what percentage of features is used for building each tree,
- reg_alpha − L1 regularization term on weights.

Each model building 50 trials, using Optuna were performed to find the best hyperparameters configuration. Each time, every parameter was changed, in given range, and used for building model on training set. After model is built, the accuracy is determined based on validation data. A program saves the best results, with the best hyperparameters values configurations. When 50 trials, were done the parameters, for the best trial (the one with the highest precision [40]), were used to build final model. Additionally, they were saved to a separate file, in case it would be necessary to reproduce models anytime.

The novel collision model is designed to predict the velocity vectors of particles after a collision based on the velocity vectors of the colliding particles. Initially, it was attempted to create a single model predicting all three velocity vectors components at once. However, despite it seems to be the most intuitive solution, such approach did not produce adequate results (in this particular case, the smallest MAE value). Therefore, targets were split, i.e. separate models were created, each for one target variable. It allowed for more accurate predictions of individual values, which is important in this context, while not affecting the performance of the model during the solution process. The final MAEs for the subsequent models used for both investigated particle sizes were equal to $\mathrm{MAE}_{u,v,w}$ =(0.5877;0.5802;0.5531) for $\mu = 315$ μm and $\mathrm{MAE}_{u,v,w}$ =(0.7078;0.7276;0.6078) for $\mu = 475$ μm.

An important element when building models, in addition to, of course, monitoring the accuracy of the model itself, is the visualization of the results. With the graphs, the modeler is able to confirm the results of the determined results, or to catch possible errors that would not be visible without such visualization. Therefore, during these calculations, graphs were made for the test set, with the true values on the X-axis and the predicted values on the Y-axis. In addition, a red line marked the straight line created based on the true values. Ideally, the true vs. predicted values would created that line. Unfortunately, when building predictive models there is always a mistake made, so the points marked

in blue should oscillate around the red line. The mentioned graphs are presented in Figure 5.5, and were created only for the test data set. The predicted values oscillate around the true values, and there are no outliers. The bar graph 5.6 shows that most of the tested configurations return the predicted velocity with an error lower than 5%.

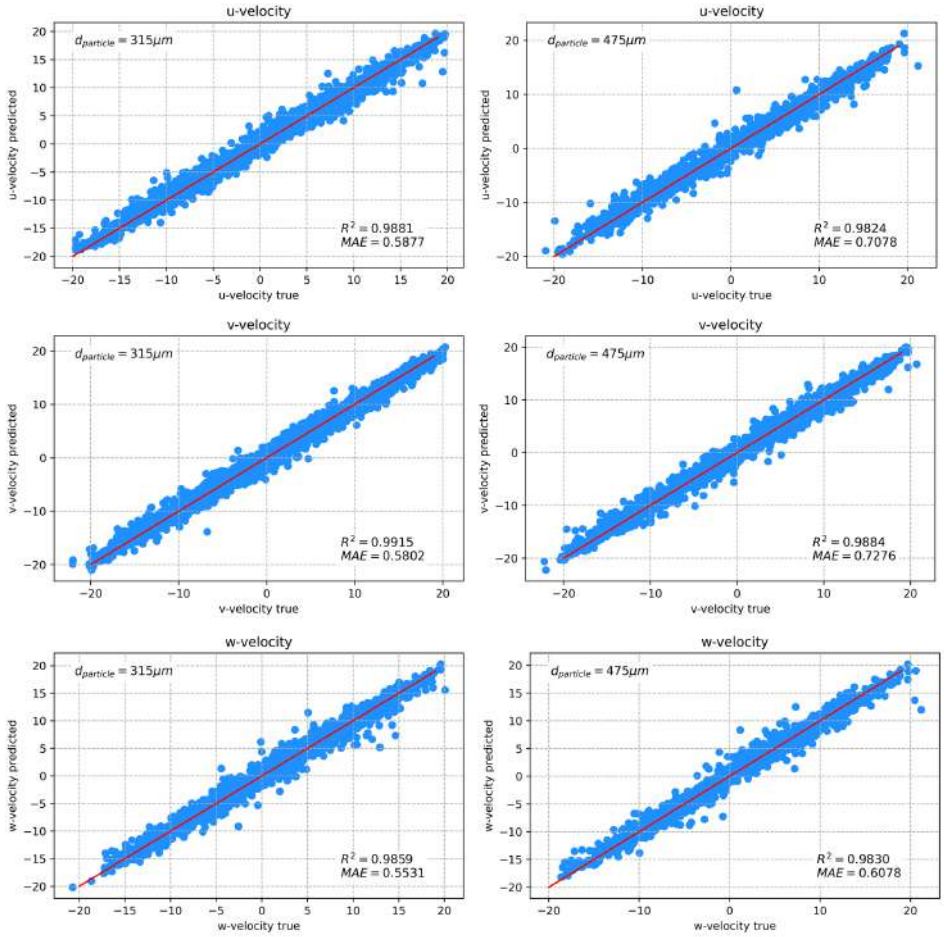

Fig. 5.5. The comparison of predicted and true values, for each of the three velocity components (**u**, **v**, **w**) for models with particle mean diameter equal to 315μm and 475μm. The red line represents only true values. The accuracy of the predictions is determined by the mean absolute error (MAE) for the test set, and marked at the bottom right corner of each graph respectively

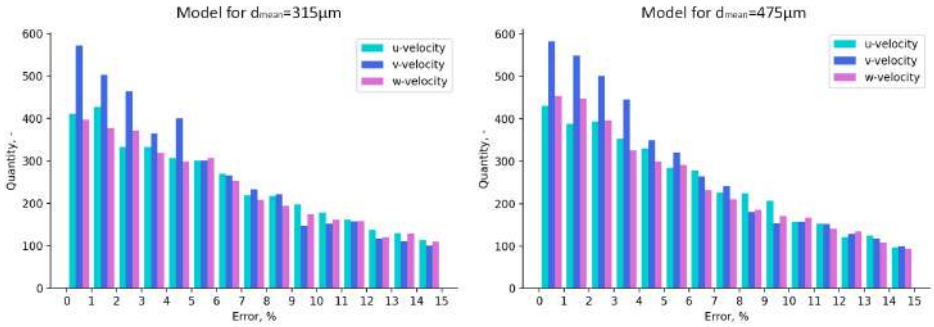

Fig. 5.6. Distribution of the calculated relative error between exact and predicted velocities for both investigated particle sizes, i.e. $\mu = 315$ μm (left) and $\mu = 475$ μm (right)

5.2.1. Surrogate collision model implementation

After the surrogate model was created, it was necessary to integrate it into the computational fluid dynamics (CFD) code. In the presented studies, the ANSYS®Fluent CFD code was used, but the model can also be implemented into other open source CFD software, like OpenFoam. As the result, the hybrid Eulerian-Lagrangian surrogate collision model (HELSCM) is obtained by modification of the standard hybrid Eulerian-Lagrangian (HEL), as was described in section 3. The general idea of the implementation is visualized in Figure 5.7. In creating the new model, the inspiration was the approach used in the DEM model. The reason is, the possibility of how to compare models similar to each other. Of course, taking into account simplifications so as not to over complicate the whole process. Collisions between particles are detected based on the distance of their particles from each other, at the end of each time step. If the distance between the centers was less than the sum of their radii ($|P_A P_B| < r_A + r_B$), the collision between the particles was detected. Accordingly, their positions and components of velocity vectors were saved. These values were stored, in case one particle was involved in more than one collision, at the end of the search and collision detection, the sum of the velocity components was divided by the number of collisions the particle was involved in. In this way, each interaction was taken into account. After the determination of the average velocities (for pairs of colliding particles), the surrogate regression model returns the values of the velocity components, after the collision, as illustrated in Figure 5.7.

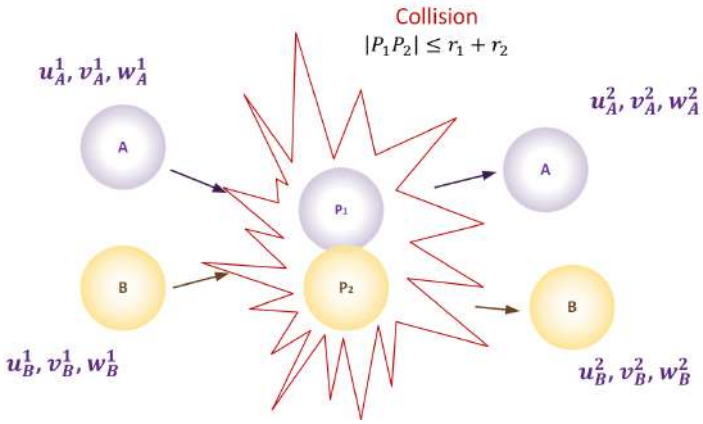

Fig. 5.7. Integration of the Surrogate model developed in the CFD code, where P_1 and P_2 are the particle centers, r_1 and r_2 the particle radii, and u, v, w are the velocity vector components of particles A and B, before and after collision

5.3. Geometrical model and mesh of the experimental test-rig

Before proceeding with the simulations themselves, it is necessary to define certain conditions, i.e. geometry, boundary conditions, initial state, and material properties, must be known at the early stage. The 3d geometry was created based the experimental test-rig, described in the section 4.1. Its main elements, i.e. the core and the side inlets, were accurately reproduced. Significant simplification was applied by removing the particle containers. To replace them, while keeping the geometry simple, the side arms were lengthened. Before making the final decision of the lenght od the extension, four different lengths of injection volumes 122 mm, 135 mm, 145 mm, and 160 mm were investigated. For each simulation, the boundary conditions and the physical properties of the particles, as well as their number, were kept the same. The average velocities of the particles at specific locations in the computational domain, were then compared between the different simulations. The results are presented in Figure 5.8. Except the extension equal to 122 mm, there is no significant difference between the velocities of the particles. According to those results, as the final length of extension, the 145 mm was chosen. The greater lengths would increase the mesh element number, and as a result the computational time. With the precised extension length, an additional volume was created, which made it possible to determine where a certain number of particles, with certain physical properties were injected.

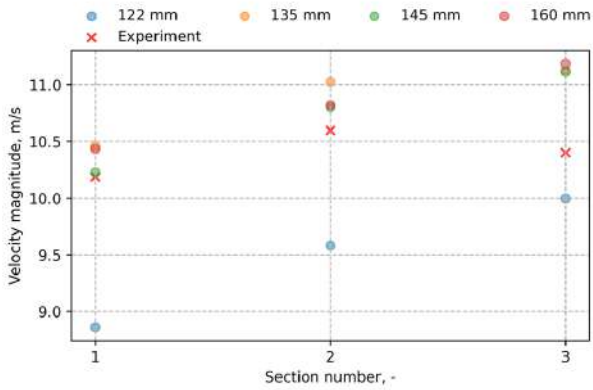

Fig. 5.8. Particle velocities determined in the three following sections, for different arm lengths juxtaposed with the velocity values obtained from the experiment

The mass of the particles, as well as their number, is determined by the amount of material involved in the experimental measurements. The position of the particles is random, in a certain volume, with the assumption that the particles do not overlap, when they enter the domain. Such way of introducing particles, allowed the best possible representation of measurements. Since the air flow fed by the side arm, it was essentially split into two parts (one flowing directly into the side channel and the other flowing through the particle container) it was necessary, to determine the initial particle velocity at the moment of injection. The air flowing through the tank impacted a velocity of the particles, so it had to be accounted for in the channel. To determine this velocity, a separate simulation of particle flow from the tank through the arm was performed. In this case, the tank was mapped exactly, and the particle velocity was determined at the outlet of the side arm. The value from simulations amounted to 3.4m /s. To perform simulations, the computational domain was discretized, using 107,000, hexagonal elements with a skewness coefficient of less than 0.22 (mean: $4.092 \cdot 10^{-2}$), where the mesh resolution was elements. To solve a set of transport equations, i.e. the mass, momentum and turbulent transport equations, the phase-coupled SIMPLE solver [21] was used in simulations. For the momentum equation, a second-order discretisation scheme was selected. For the continuity equation, the QUICK scheme [49] and for time discretization, the first-order implicit scheme was used. To quantitatively compare the results, ten sections, highlighted in red (Figure 5.9), were designated. Their locations correspond exactly to those in the main channel, in the experimental rig. In the given sections, particle velocities, averaged over time, were determined along the length of the section. This allowed, to create velocity profiles, which could then be quantitatively compared with each other. The boundary conditions for the side and main inlets are velocity inlet, at the outlet the pressure outlet

was set. Particles were introduced once, in the first step of the simulation at a velocity of 3.4 m/s, into the volume of the side arms. For clarity, all mentioned, boundary conditions are marked on Figure 5.9.

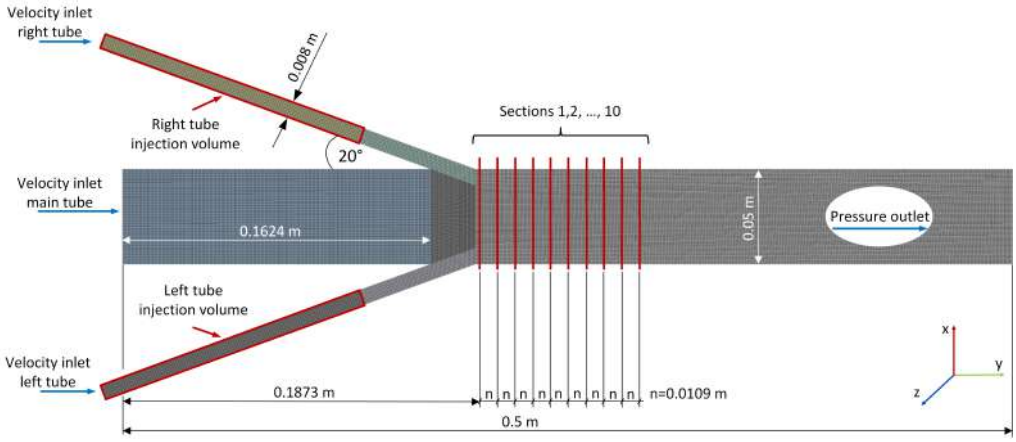

Fig. 5.9. Geometry and mesh of the computational domain, with marked dimensions and sections, where velocity profiles are determined to allow comparison of simulation results with experimental measurements

5.3.1. Comparison of a simulation results using the DEM and HEL approaches

Before comparing the functionality and determining the correctness of HELSCM, a comparison of the simulation results using already available (i.e. DEM and HEL) was made. To compare the two models, numerical simulations were performed for three different time steps, i.e. $\Delta t = 10^{-3}$ s, 10^{-4} s and $7.14 \cdot 10^{-5}$ s. The biggest time step size is the one, most often used when applying the HEL approach [15], while the smallest time step size corresponds to the frequency of recorded images, during experimental work [140]. Three different time steps were chosen to analyze their effect on particle behavior in both models. In addition, using the HEL model, two configurations of simulations were performed. In first case, common for DEM and HEL, 12,000 particles (with a total mass equal to 0.000196 kg), where one parcel corresponded to one particle, were injected. In second configuration, only for HEL simulations, the mass of injected material was increased to 0.0018 kg. Such modification meant that, while keeping the same number of parcels introduced, the number of particles in them increased (up to 8), affecting the diameter of the plots itself. The effect, was to increase the volume of the solid phase in the computational cells.

The purpose of these simulations was to check the impact of the time step on collision detection by both models. To model the collision by application of the HEL approach, using the KTGF set of closure terms that need to be modeled. The list of used submodels is presented in Table 5.1.

Table 5.1

Closure models and parameters used in the simulations, where model names are given as they appear in ANSYS®Fluent CFD code

Solid pressure, p_s	lun-et-al [85]
Radial distribution, $g_{0,ss}$	lun-et-al [103]
Granular viscosity, μ_s	gidaspow [58]
Granular bulk viscosity, λ_s	lun-et-al [85]
Frictional viscosity, μ_{fric}	schaeffer [116]
Frictional pressure, p_{fric}	based-ktgf [124]
Granular temperature, Θ	algebraic [124]
Drag model, K_{sg}	gidaspow [58]

One key distinction between the DEM and the HEL is the way, in which they handle interparticle collisions. To compare the results, the particle positions and the contours of the volume solid fraction, in 0.03 s of the simulation were compared with each other, as illustrated in Figure 5.10). Due to collision detection, based on their location in the computational domain the DEM model is more effective, as the interactions between particles are clearly visible. When the two streams meet in the main channel, they interact with each other and their trajectories are altered. It results in stream mixing and moving along the core of the channel. The HEL model, which uses information about solid volume fraction in computational cell and calculated granular temperature, does not precisely detects particle collisions. Two opposite streams, when meeting in the core of the canal interpenetrate each other. Such issue is also similar, when the volume fraction is increased (i.e. in case, where one parcel contains few particles). Precise detection of collisions between particles in the computational domain is a very important aspect, in simulations, where high computational accuracy, reliability, and in situations where reality must be reproduced as accurately as possible. An attempt was therefore made, to develop a model that, at best, combined the advantages of both models. And the result, would be the ability to detect as many collisions as possible, in a reliable manner.

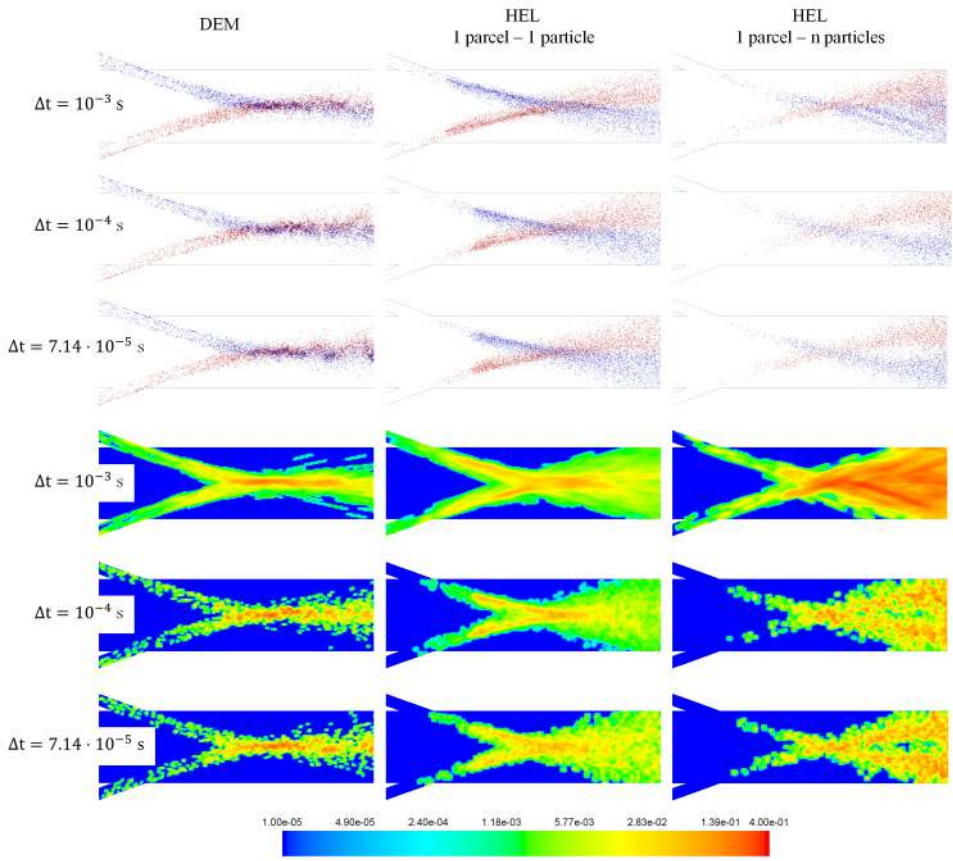

Fig. 5.10. A qualitative comparison of the results of simulating the collision of two streams in a main channel. The first three rows, show the positions of the particles, with the right stream in red and the left stream in blue. The following three rows, show the solid volume fraction in each computational cell. The HEL model was used to analyze two different scenarios. In the first case, each parcel contained one particle, while in the second case, each parcel contained n particles

One of the drawbacks, concerning the DEM model, is the calculation time, therefore the comparison was made between the calculation time for the DEM and HEL models using a chart, presented in Figure 5.11. Definitely using DEM, the time required for simulations might be up to 28 times longer than for HEL. The simulations in this study, involved a flow with 12,000 particles, but in industrial applications, the number of particles can be up to several millions. It is the reason that precludes the use of the DEM model in industrial calculations. The waiting time, could last for even years, which in a rapidly developing world and technologies is an unacceptable period. Using the HEL model, the calculation time is much shorter. However, a factor that is also a disadvantage

is the omission of collisions that occur, in case the solid fraction is too small. Such factor also makes the model sensitive to the mesh size, which further has the effect of increasing volume fractions, which consequently leads to clustering of particles [134]. The implementation of the HEL approach, for drag calculation in motion equation, uses the diameter of the parcel instead of the diameter of the particles, which also affects the sensitivity of the HEL approach. Therefore, to maintain the speed of the HEL, the new approach will be developed. The working principle of the HEL approach will be preserved, but the collision term, as well as the collision detection will be replaced with surrogate collision model (SCM). Data collected, from particle collision simulations using the DEM model, will be used as the basis for the SCM building.

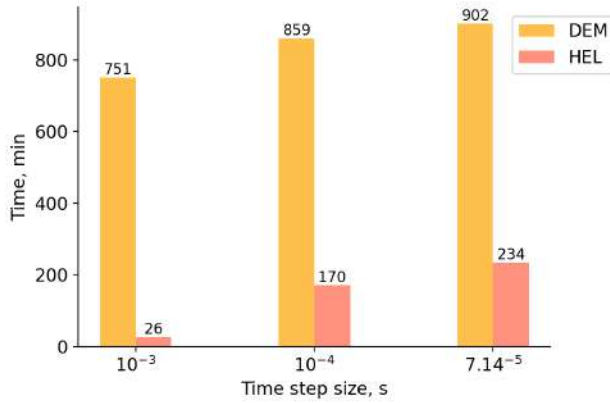

Fig. 5.11. Comparison of DEM and HEL simulation times

5.4. HELSCM technique validation

Before testing on real objects with many interactions between particles, which can be difficult to control, tests were first conducted on a small scale for two groups of colliding particles, as illustrated in Figure 5.12. The physical properties of the particles, coincide in those used in experimental measurements. An additional assumption was also made, to determine the small volume fraction of the solid phase in computational cell, to verify whether the most important disadvantage of the HEL model can be eliminated by using a novel approach. Figure 5.12 The results from the three models, DEM, HEL and HELSCM, were compared. Particles were introduced into the computational domain with an initial velocity of 5 m/s, at an angle of 20 degrees. The chosen angle for low-level testing, results from the angle of the arms to the main channel, in the final geometry. This angle, is marked in Figure 5.9. Additionally, for clarity Figure 5.12, also has marked the collision angle with

the direction of movement of particles. Their positions were captured before, at and after the collision. The simulation time step was equal to 10^{-4} s. The initial positions of the particles were accurately determined, according to the real trajectory of the particles, so that at the fifth time step the particles did definitely collide with each other. Thus, the behavior of particles on collision can be accurately observed. The HEL model did not detect collisions between particles; at the time of the collision, the particles overlapped, without interaction, as can be seen in the illustration showing the position of the particles after the collision. The solid volume fraction was smaller than 0.06, which further proves the model's failure to detect collisions at low contribution. Using the DEM model, a collision between two streams is detected. At the time of the collision, the particles do not overlap, but after the collision, a change in the direction of the particles is observable. Their trajectory changes, the particles bounce at the same angle at which they collided. In the case of the HELSCM model, the particles overlap when they collide (similar to the HEL model). However, at this point, a collision is detected (the distance between their centers is less than the sum of their radii) so a simplified model is invoked, and returns the particle velocity components after the collision. Consequently, the particles change their trajectory.

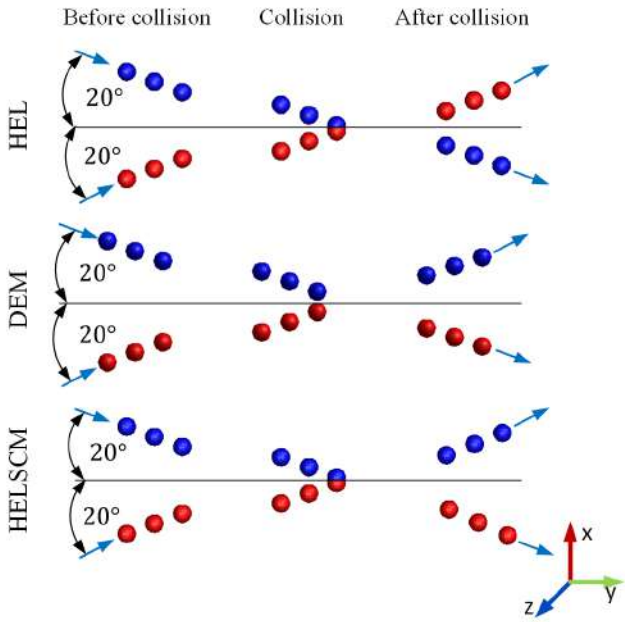

Fig. 5.12. Comparison of collisions of two particle streams, for three different models, made for small-scale simulations to confirm the functionality of the newly developed model. Position of particles are captured before, at the moment and after the collision

5.5. Lab-scale application of HELSCM collision model

The following step after introducing and verifying the accuracy of the HELSCM model in micro scale, was to use it for simulations reflecting the particle flow in the experimental test-rig. The results obtained were then compared with results from simulations, using the DEM model and measurements. As in the case of calculations using, already available models, exactly the same geometry was used, illustrated in Figure 5.3. Several configurations of calculations were carried out. Collisions of two particle streams were investigated, where their mean diameters were equal to 315μm and 475μm. In addition, for each diameter, three different time steps were used, i.e. $\Delta t = 10^{-3}$ s, 10^{-4} s and $7.14 \cdot 10^{-5}$ s. Particles were introduced in the first time step using volumetric injection; the exact volumes into which the particles were introduced, are marked in Figure 5.9. For simulations with an average particle diameter of 315μm, the number of particles injected was 12,000, while for larger diameters it was 3600. The difference in the number of particles injected is due to the constant mass of the material, which was equal to 0.000196 kg.

To compare the velocity profiles quantitatively, the velocity of particles was determined measured at 10 sections along the length of the main channel. In addition, the sections were divided lengthwise into 20, 85, 125 elements. The number of boxes, affects the accuracy of the average velocity values and how closely the velocity profiles reflect the true shape. The more elements there are, the more accurate the results are. Especially, in the first three sections, where there is a so-called dead zone. It means that in the center of the main core, in the first three sections, the particles only leave the side arms and there are no collisions between opposite streams. This has a significant impact on the determination of velocity profiles using 20 boxes, where the lack of collisions in the early stages of the flow is imperceptible. For more boxes, the results are not significantly different.

As the simulation time step increases, the HELSCM model, deviates from the measured results as well as the results from the DEM and HEL models. As the time step decreases, more collisions are detected, significantly increasing the divergence and inter-particle interactions (Figure 5.16). In addition, dispersion can be observed in the pictures at the last rows (Figure 5.16), representing the volume share of the solid phase, where a logarithmic scale was used to better illustrate small fractions. The results were presented in 0.02 second of simulation. The current implementation of the SCM in the HEL approach does not consider the duration of the particle collision. Using a small time step can result in multiple detection of the same colliding particles, which can perturb their predicted velocities and trajectories due to the repeated calls to the collision model. This can affect the dispersion of the particles in the computational domain (Figure 5.16, images presenting the position of particles).

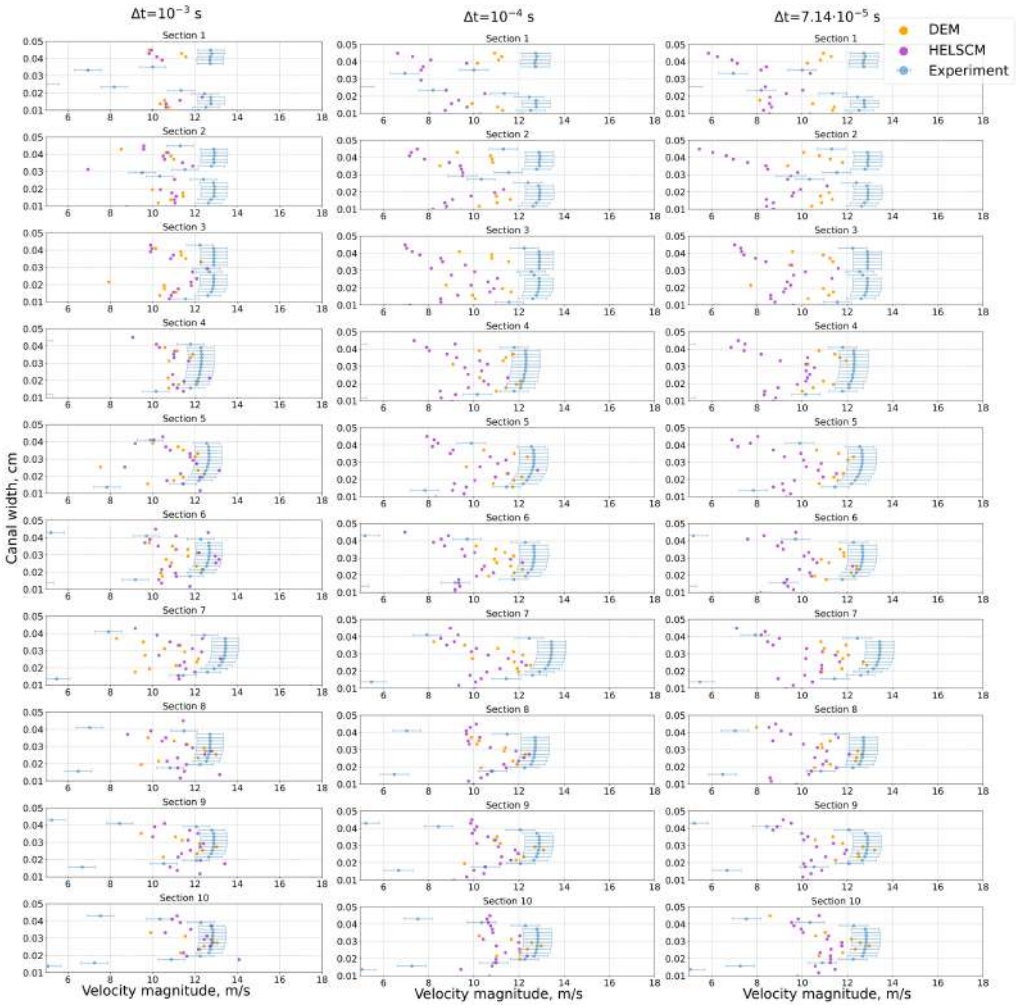

Fig. 5.13. A quantitative comparison of the DEM and HELSCM models, made based on the calculated velocity profiles of particles mean diameter equal to 315 μm, at ten locations set in the computational domain (Figure 5.3) after 0.02 seconds of simulation using 20 boxes for profile determination. The comparison was made for three different time steps.

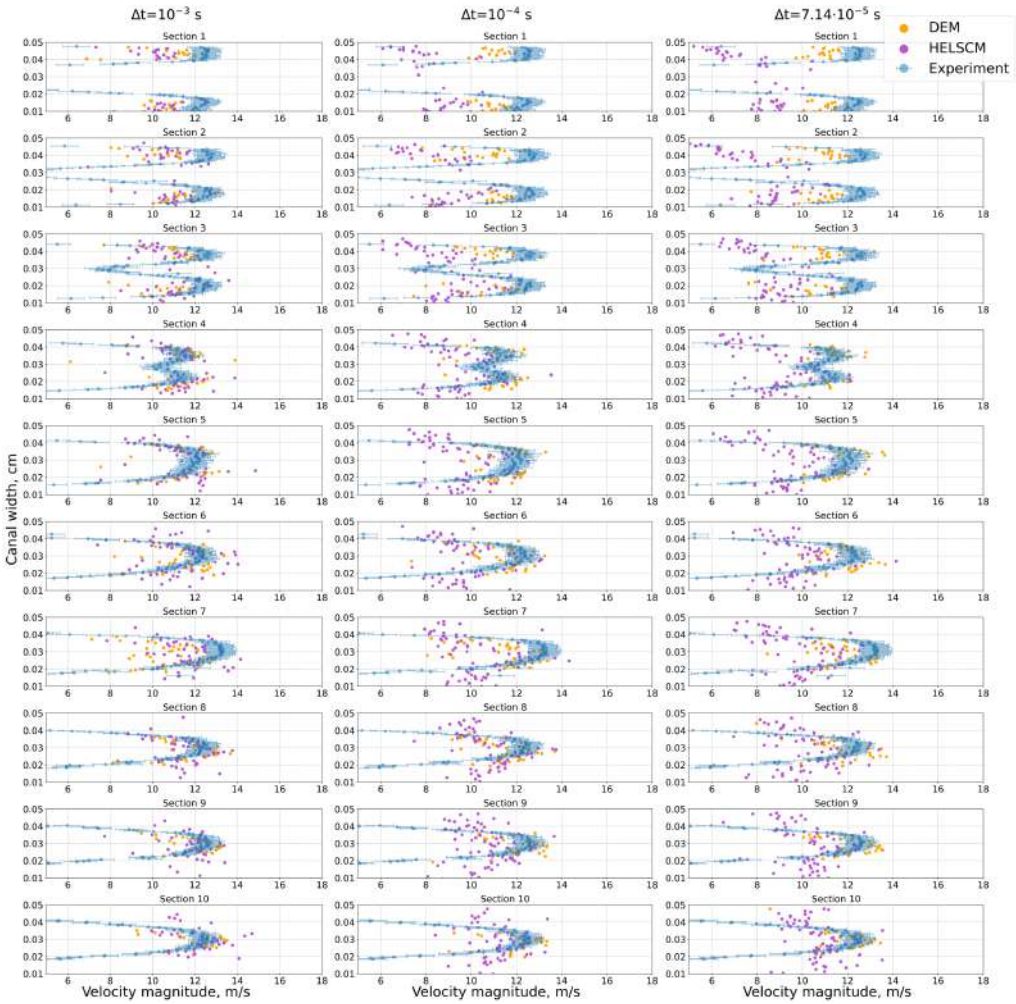

Fig. 5.14. A quantitative comparison of the DEM and HELSCM models, made based on the calculated velocity profiles of particles mean diameter equal to 315 μm, at ten locations set in the computational domain (Figure 5.3) after 0.02 seconds of simulation using 88 boxes for profile determination. The comparison was made for three different time steps.

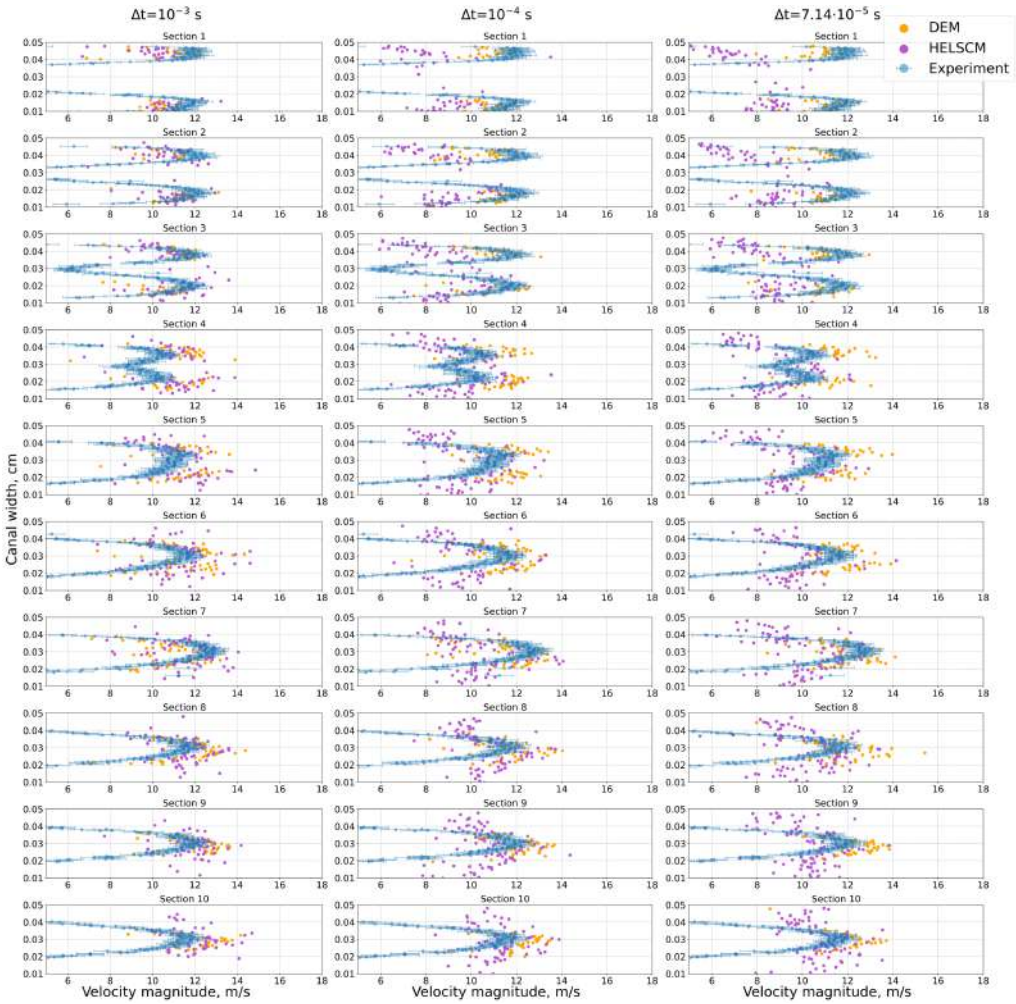

Fig. 5.15. A quantitative comparison of the DEM and HELSCM models, made based on the calculated velocity profiles of particles mean diameter equal to 315 μm, at ten locations set in the computational domain (Figure 5.3) after 0.02 seconds of simulation using 125 boxes for profile determination. The comparison was made for three different time steps.

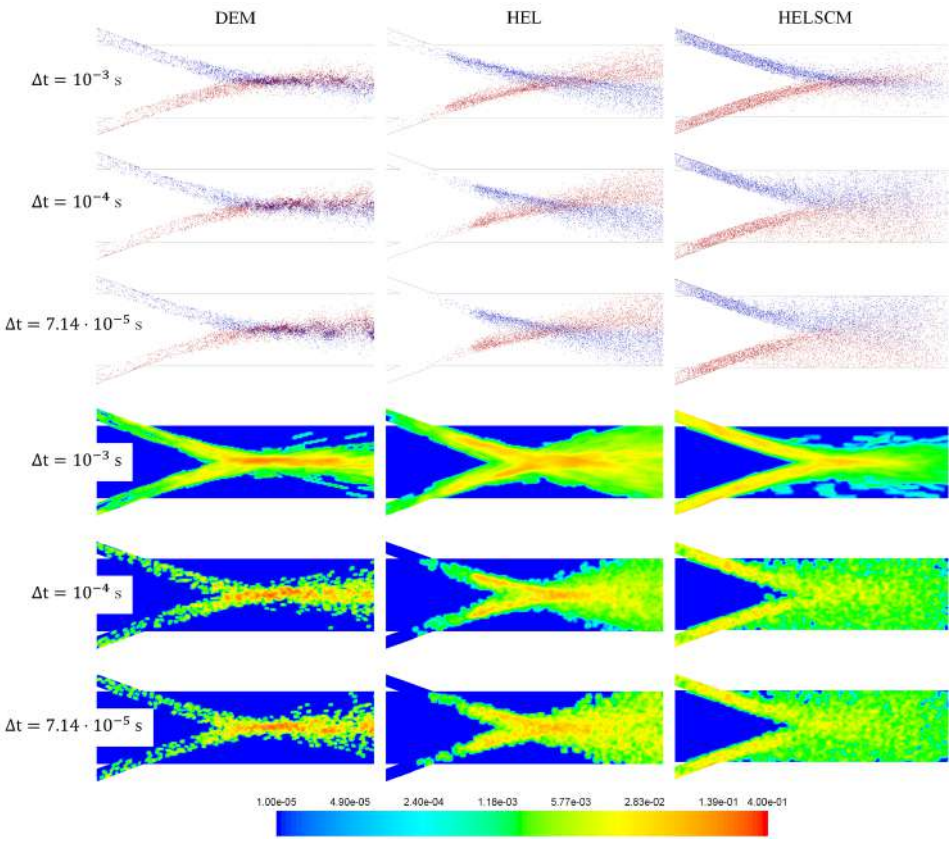

Fig. 5.16. A qualitative comparison of the results of simulating the collision of two streams in a main channel. The comparison is made between results of three models DEM, HEL and HELSCM. The first three rows, show the positions of the particles, with the right stream in red and the left stream in blue. The following three rows, show the solid volume fraction in each computational cell

To optimize the performance of the HELSCM model, it would be beneficial to monitor the rate at which particles collide with one another. However, doing so may slow down the overall execution time of the model. Increasing the accuracy, by adding more elements involved in the model's operation, could lead to a definite increase in calculation time. Such an effect, would make the newly created model temporally close to the DEM model. During the development of the HELSCM model, one of the main objectives was to reduce computation time. The minor differences between experimental and numerical results are observed for simulation time step equal to 10^{-3} s. This pattern can be seen in the results from both numerical models at all ten locations. Using DEM model, the impact of particle collisions is reduced, causing particles to alter their path and continue moving in the same direction along the channel rather than scattering after collision (presented

in Figure 5.16). Therefore, particle velocity is higher in the center of the main channel, which is proved on velocity profiles graphs in Figure 5.15.

To compare the results with each other qualitatively, the recorded results juxtaposed at the same time (t=0.03 s) in Figure. 5.17. Using the smallest time step with both numerical approaches, the experimental particle distribution falls between the results obtained with the DEM and HELSCM models.

The most similar results, are those, where the largest time step was used in the simulations (i.e. 0.001 s). The summary in Figure 5.15), includes the results of simulations using the HEL approach, which, regardless of the time step used, does not detect collisions between particles that are part of opposite streams. The particles penetrate each other. Influencing it, is the low volumetric contribution of the solid phase to the cells volume, as shown by its contours (Figure 5.16). That's why, despite the visually strong similarities at first glance, the HEL model does not meet expectations for collision detection. Comparing the results obtained, both qualitative and quantitative, the assumption of the HELSCM model was upheld. By replacing the component responsible for detecting collisions (based on the determination of granular temperature) with a machine learning algorithm, the number of collisions detected was increased, while keeping the execution time relatively short. Relying on the Figure 5.21, it is clear that the simulation time using the HELSCM model is significantly shorter compared to the DEM model. When compared to the simulation time using the HEL model, the time is slightly longer. The situation repeats for each of the three, analyzed time steps.

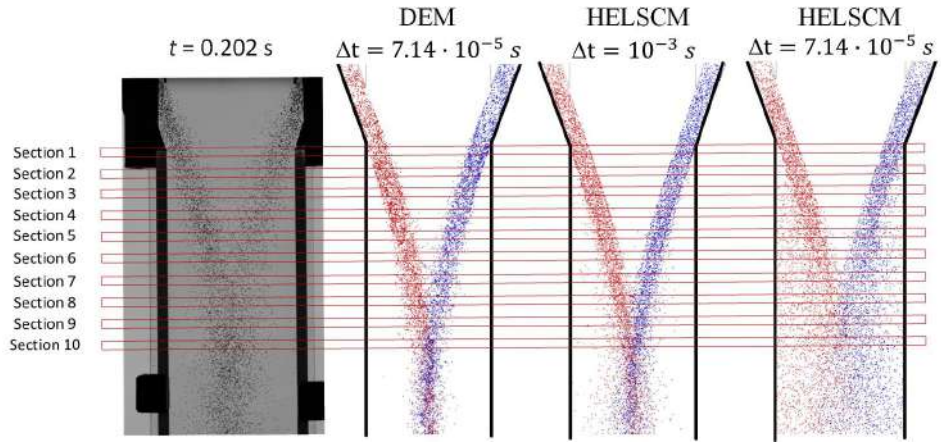

Fig. 5.17. Comparison of numerical results with experimental data based on particle dispersion in the computational domain determined after 0.03 seconds of wall clock time of numerical simulations using DEM and developed HELSCM approaches for particle size 315 μm and marked locations used for the determination of particle velocity profiles.

For analysis of the simulations, with particles mean diameter diameter of 475μm, the velocity profiles were compared only using the highest number of boxes, which was 125. The comparison was conducted at three different time steps, as in previous cases (Figure 5.18).

The largest differences are most apparent in the final sections. The most similar results, and preserved velocity profiles, are for the largest time step, which is similar behavior to simulations using particles with smaller diameter (315μm). Shifting of two streams, results in moving the collision region closer to the inlets. The effect is definitely greater dispersion of particles just after the outlet of the side channels, which is noticeable in the graphs of velocity profiles, for the first sections (Figure 5.18). Analyzing the records of measurements, in the form of images, one by one, it was observed that there happen to be single particles that, due to, for example, collisions with walls, change their trajectory to perpendicular to the main channel axis, or move backwards. Both for the case analyzed, where the average diameter was 315μm, and for the case with an average particle diameter of 475μm, the determined Stokes number (St= t_F/t_{RT}) for a given flow condition was greater than the unit. The characteristic time of the flow field (t_F) was determined by considering the dependency between the area of the channel cross section, its perimeter and the mean velocity of an air. Response time (t_{RT}), strictly depends on the diameter of the particles that are part of the flow, which for mean particle diameter equal to 475 μm (t_{RT} = 0.25 s) was almost twice as high compared to the response time calculated for particle mean diameter 315 μm (t_{RT} = 0.13 s) The mentioned particle response time, was determined using the simplified equation of motion (2.26) for a single particle (the terms ∇/ρ_p and F_{coll} are neglected). High values of the Stokes number (St) cause the time required to stabilize the flow to increase, in case of sudden changes in the direction of particle movement. The algorithm, used in presented work, described in subsection 4.2, only tracks the movement of particles flowing downstream and having an impact on the calculated velocity profiles.

Particle positions and contours of solid phase volume fraction, for calculations performed with particles with an average diameter of 475 μm, are presented in 0.02 s of calculations and summarized in Figure 5.19. In DEM application, the movement of particles was mainly monitored in the center of the main channel and observed over the course of the three time steps studied. Comparison the experimental results with the simulation results, presented in Figure 5.20, using the DEM model, the particles in the core itself are close to each other and move in one direction along the channel. Therefore, for the largest time step, the HELSCM model, visually, is much closer to the experimental measurements, which is influenced by the subtle dispersion of the particles before and after the collision. For the smallest time step, the particles definitely interact more with each other and propagate through the channel.

Analyzing Figures 5.19, the conclusion is that for the specific configuration of the HEL model and the solid volume fraction used, the collision effect was not visible for either particle diameter. Additionally, the HEL approach is not affected by the size of the particles.

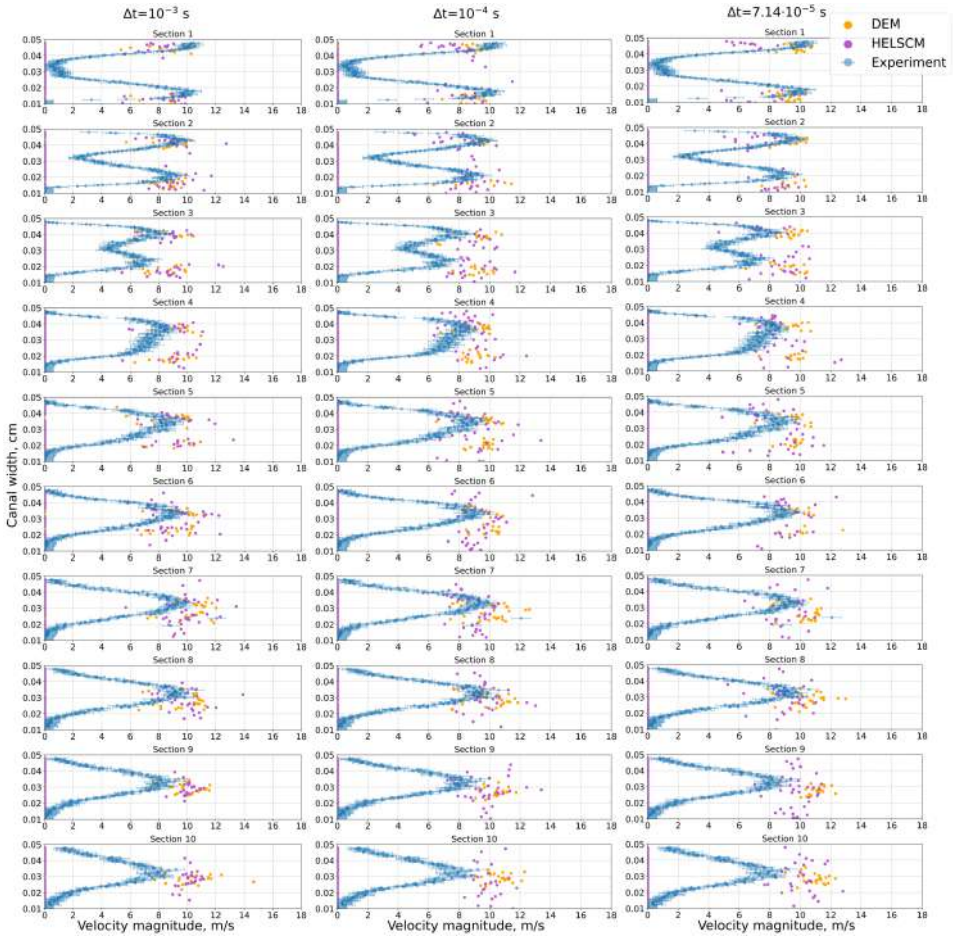

Fig. 5.18. A quantitative comparison of the DEM and HELSCM models, made based on the calculated velocity profiles of particles mean diameter equal to 475 μm, at ten locations set in the computational domain (refer to Figure 5.3) after 0.02 seconds of simulation using 125 boxes for profile determination. The comparison was made for three different time steps.

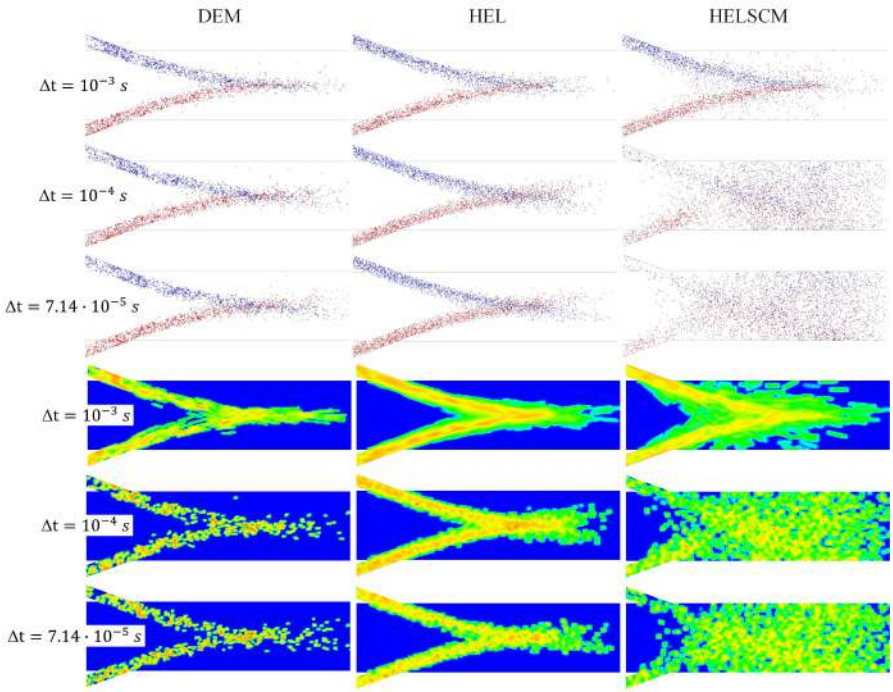

Fig. 5.19. Qualitative comparison of the results of simulating the collision of two streams in a main channel. The comparison is made between results of three models DEM, HEL and HELSCM, for particle mean diameter equal to 475 μm. The first three rows, show the positions of the particles, with the right stream in red and the left stream in blue. The following three rows, show the solid volume fraction in each computational cell

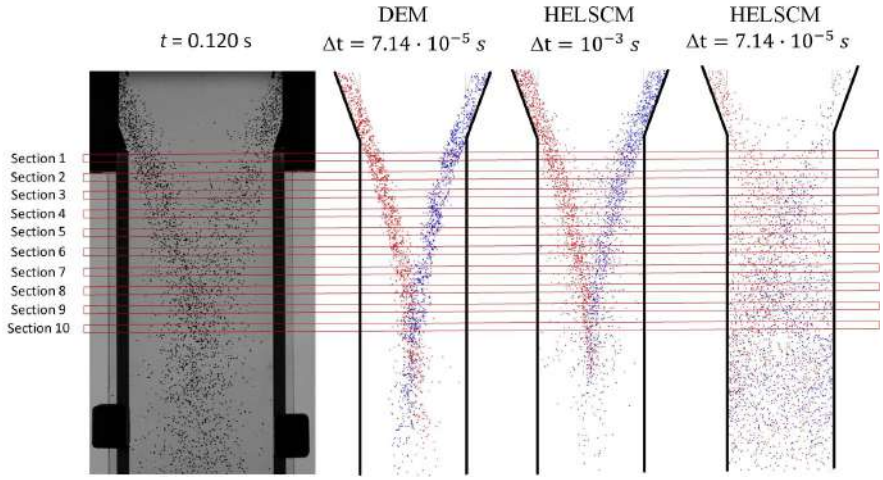

Fig. 5.20. Comparison of numerical results with experimental data based on particle dispersion in the computational domain determined after 0.02 seconds of wall clock time of numerical simulations using DEM and developed HELSCM approaches for particle size 475 μm and marked sections used for the determination of particle velocity profiles.

For each of the cases discussed, i.e. for the two different mean particle diameters (315 μm and 475 μm), the three numerical models used (DEM, HEL, HELSCM) and the three different time steps ($\Delta t = 10^{-3}$ s, 10^{-4} s and $7.14 \cdot 10^{-5}$ s), the time required for the same number of iterations was measured, assuming the same initial conditions. In addition, the calculations were performed on the same machine, using one core. The results were collected in the form of a graph, presented in Figure 5.21. By comparing the times, it is apparent that the time required, to perform the same simulations, is significantly shorter than that required for the DEM model. This time is close to the time that must be spent on calculations using the HEL model. The time for simulations, where the average diameter was 475 μm was lower than for smaller diameters, due to the smaller number of particles involved in the flow (it is the effect of maintaining the same mass of material in the domain). During the simulation, not only time was measured, but also the number of collisions detected, which directly correlates with the number of times the simplified model was called. An additional factor contributing to the shorter time is the fewer times the simplified model was called.

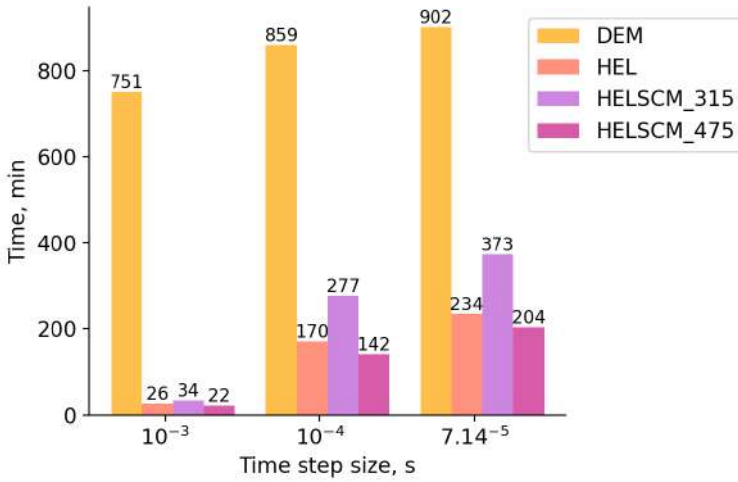

Fig. 5.21. Comparison of DEM and HELSCM execution times, carried out using a single process at the same work station

Another point of comparison involved observing the frequency of collisions detected during each time interval. The number of collisions was counted for simulations with two different diameters and three different time steps. In both cases of diameter, the highest frequency of collisions occurred when using the shortest time interval. Similarly, when employing the longest time interval, the recorded instances of collisions were at their lowest. This pattern aligns with the visual analysis of the results presented in Figures 5.17

and 5.20. In instances of the shortest time interval, the scattering of particles within the domain is most conspicuous due to the prevalent collisions that take place and are identified. Conversely, the utilization of the longest time interval, resulting in the fewest collisions, leads to minimal particle dispersion.

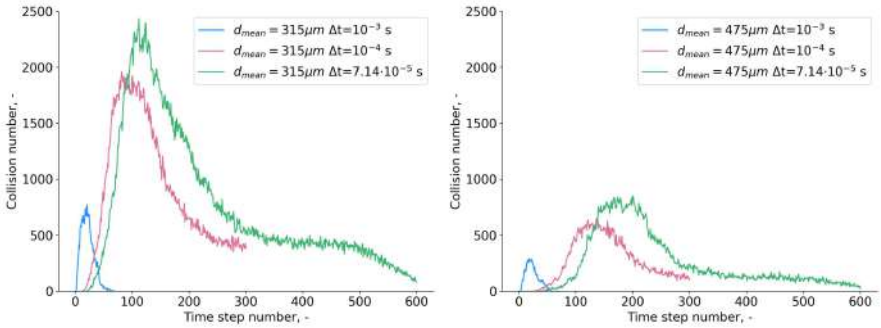

Fig. 5.22. Comparison of collision rates calculated for three investigated time steps of numerical simulations and two used particle diameter

5.6. Testing developed collision approach for simplified CFB unit

Earlier tests, collisions between two streams of particles, were a prelude to testing the feasibility of modeling the fluidization process. The smaller scale, and far fewer collisions, allowed for a more accurate analysis of the collisions, using HELSCM. Therefore, this step was necessary before, simulations in a flow, with more collisions were attempted. The process is much more complex. A circulating fluidized bed boiler was chosen, as the test model. In this process, the fluidization speed is adjusted, so that the layer at the bottom of the boiler fluidizes, but is high enough that some of the material is returned through cyclones, back into the boiler. Thus, the amount of mass in the boiler is preserved. It is also worth noting, that the tests are carried out in the so-called cold state. Only the flow of material is modeled, without taking into account the heat transfer and combustion process. Modeling of these phenomena, is not the subject of this paper, but for the curious reader, an example of the description of such simulations is the work of [14].

5.6.1. Simplified geometry of the circulating fluidized bed boiler and recirculation procedure

The geometry of the boiler was imitated, in a very simplified manner. The main reason was, as in the case of particle collisions in the canal, less particles facilitates to

have a control over the process, and so that it would be easier to detect the reasons for any inaccuracies. The shape of the numerical model used in calculations, closely resembles the physical design of the boiler. The model has a rectangular top and sloping walls at the bottom, with the base being smaller than the outlet. There are also two inlets located on the sides. The geometrical mesh consists of 13,300 elements, with the skewnees coefficient lower than 0.76606 (average value equals to 0.020583). To solve a set of transport equations, i.e. the mass, momentum and turbulent transport equations, the phase-coupled SIMPLE solver [21] was used in simulations. For the momentum equation, a first-order discretisation scheme was selected. For the continuity equation, the QUICK scheme [49] and for time discretization, the first-order implicit scheme was used. Simulations were carried out using only the HEL and HELSCM models.

To compare the results quantitatively, 18 planes were created along the boiler height, perpendicular to the geometry axis. The planes served as a location, determining time-averaged pressure values. The geometry used, along with the marked grid elements and applied boundary conditions, is presented in Figure 5.23A. In addition Figure 5.23A, the volume in which the particles were introduced is marked. For readability, Figure 5.23B, includes a schematic of the geometry with the dimensions marked, and an example of one plane from which the pressures were read, in order to compare the results between models. Due to the simplified geometry, but the need to replicate the circulation bed as closely as possible, an important step was to ensure recirculation of the material. A User Defined Function was used for this purpose. The principle of its operation is simple, nevertheless the implementation was not one of the easiest tasks. The function's duty is to sum the mass and number of particles leaving the computational domain at the end of a given time step. This value is retained in memory until the beginning of the next time step. In the next step, the determined number of particles of a given diameter is entered into the designated volume. Maintaining a certain number of particles leaving the computational domain and returning was a challenge. But by keeping information about the specific number of parcels leaving the domain, it was possible to set a specific number to be returned. By default, the entire mass is divided by the number of cells present in the given volume, and as many particles as elements are thrown in. Therefore, in each iteration, the number of particles would increase as well as their diameter would decrease. Which would not reflect the occurring process accurately enough. This is a solution that worked very well in this case. However, one limitation should be mentioned, namely, the number of cells in the given volume must be large enough to drop each particle (since only one particle is dropped into one element). Otherwise, not all the mass will be returned. In the case currently under discussion, i.e. simulation of particle collisions in a circulating fluidized bed, the place of particle injection is the volume, nevertheless this choice is up to the user and can also be, for example, the surface. The principle of operation is also

illustrated in Figure 5.23C. Marked are the area, where the particles were caught, at the end of a given time step, and some points, where they were returned in the next step. Also marked are sample planes, in the cross section of the computational domain, where average pressures were determined. The values, were used to compare the pressure drop between models. The diagram only shows the cells present in an illustrative manner, and their size is exaggerated for clarity.

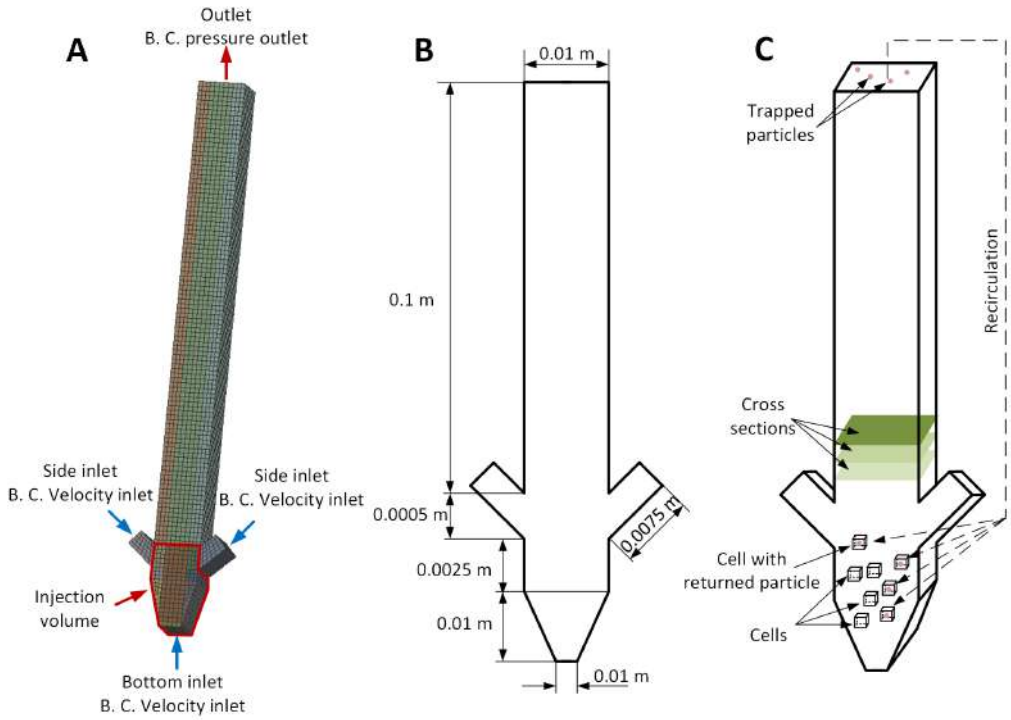

Fig. 5.23. **A:** Geometry of a simplified CFB boiler with boundary conditions and injection volume. **B:** Schematic of the boiler, with dimensions marked. **C** 3D diagram with marked cells, particle capture location and their return point (in this case into the certain computational cell) as well as the sample planes, where average pressure was read

5.6.2. Recalibration of the surrogate model

The developed approach to data generation, used to create the simplified model is universal. I.e., every effort was made to ensure, the algorithm for the generation of collision configurations, covers them throughout 3d space, so collisions occur from every direction, in a variety of possible arrangements. Nevertheless, it must be remembered that, like any

model created or already available, it must be tailored to the case in which it is to be used. In this particular case, the use of the HELSCM model, to model the fluidization phenomenon, it was decided to recalibrate the model, used in the calculations of collisions between two particle streams.

The velocities of the incoming air, as well as the velocities of the particles themselves in the fluidized bed, are much smaller than in the previous experiment [111]. Consequently, having a velocity range of 0 to 20 m/s, the model predicting low values made errors, specially with lower velocities (see Fig. 5.6). Therefore, based on the approach already used, a model was created, whose speed range, it was taught, covered speeds from 0 to 2 m/s. The learning data from the first iteration and the second iteration were not combined, in order to build models (recalling, one model was built for each velocity component). Combining data, does not always benefit its accuracy. Nevertheless, in this case the goal was to best represent the low velocities that the particles have, especially at the bottom of the boiler. Therefore, two separate groups of models were used. If the components of the input velocity vectors were low (that is, they ranged from 0-2 m/s) the model, for lower velocities, was called. On the other hand, when the velocities in the input were higher, a primary model was called up, created on the basis of velocities in the higher range.

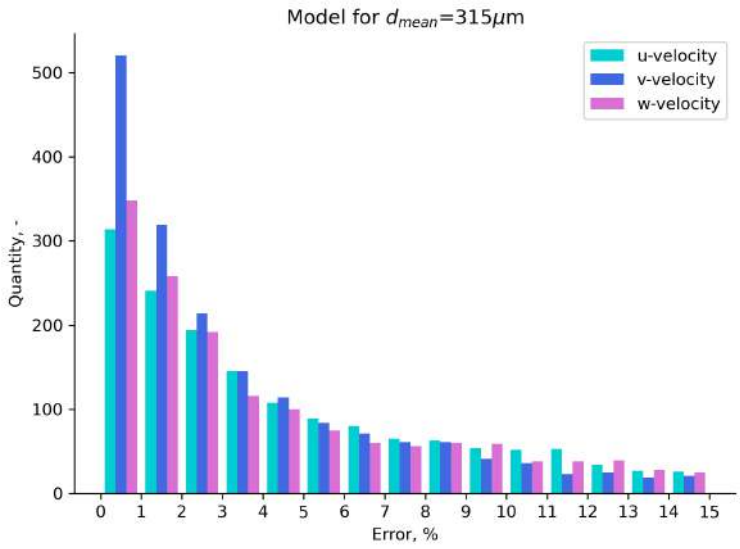

Fig. 5.24. Distribution of the calculated relative error between exact and predicted velocities for recalibrated model, with mean diameter equal to 315 µm

5.6.3. Comparison of simulation results using HEL and HELSCM approaches

Simulations, were performed for two models HEL and HELSCM. The models differ from each other, only in the way they detect collisions. To compare the two models, numerical simulations were performed for one time step, i.e. $\Delta t = 10^{-2}$ s. Simulations were performed for only one time step, since the main purpose of these calculations was to verify the feasibility of using a simplified model to model the fluidization phenomenon. Further, more detailed studies will be carried out in future work to improve the model. To model the collision by application of the HEL approach, using the KTGF set of closure terms that need to be modeled. The list of used submodels is presented in Table 6.3. For the main and side inlets, the discrete phase BC type was set to bounce, and for the outlet the particles were trapped and processed using the User Defined Function (the operation principle briefly described in subsection 5.6.1.

Table 5.2

Closure models and parameters used in the simulations, where model names are given as they appear in ANSYS®Fluent CFD code

Solid pressure, p_s	lun-et-al [85]
Radial distribution, $g_{0,ss}$	lun-et-al [103]
Granular viscosity, μ_s	gidaspow [58]
Granular bulk viscosity, λ_s	lun-et-al [85]
Frictional viscosity, μ_{fric}	schaeffer [116]
Frictional pressure, p_{fric}	based-ktgf [124]
Granular temperature, Θ	algebraic [124]
Drag model, K_{sg}	gidaspow [58]
Restitution coeff. in solid phase, e_{ss}	0.9
Friction packing limit, $\varepsilon_s^{\text{fric}}$	0.61
Packing limit, $\varepsilon_s^{\text{pk}}$	0.63
Transition factor of HEL	1.0
Particle-wall reflection coeff.	$e_{nor.}$=0.8; $e_{tan.}$=0.8

Geometry used as a whole, essentially was divided into two volumes. One of them (the smaller one, marked on red (refer to Figure 5.23A), was used to initialize the particles in the first step of the calculation. To the cells in this part, particles, which left the computational domain at the end of the time step, were returned. Such procedure was used intentionally, because in a real boiler, particles that should be returned to the boiler are introduced through the side inlets, located at the bottom. The total simulation time

was 10 s. From the 5th second, time averaging of the instantaneous values and root-mean-squares of the fluctuating values were sampled during the calculations. The above steps are common to both models (HEL and HELSCM). After the simulations, the averaged pressures, solid-phase volume contributions and particle velocities were compiled together for the results of each model. The results were combined and presented in Figure 5.25).

Both models gives similar trends. For pressure, the value is highest at the bottom, and decreases as the height increases. It aligns with reality, as there is the most of the material at the bottom, which leads to a higher pressure. The HELSCM model, even though it has a uniform distribution, has slightly higher pressure values, at the bottom, than the HEL model. The distributions of volume solid fraction are also similar. For both cases, the largest share is at the bottom of the boiler, confirming the results of the pressure. Due to the particles being returned to a certain volume, there is a negligible contribution in the side inlet channels. In HEL simulations, along the walls and in the channel core, the distribution of particles is uniform. In the case of HELSCM simulations, on the other hand, one notices a higher volume share of particles at the bottom of the boiler, especially near the walls. In addition, the volume solid fraction is also bigger, in the higher parts near the walls and lower in the core. Note that the scale determining the share is logarithmic.

The distribution of particles in the computational domain, as well as their velocities, are also similar. The velocities do not deviate from each other, the particles are evenly distributed around the boiler. Confirming the previously discussed results of pressure and volume shares, the largest number of particles is at the bottom of the boiler, and their number decreases with height. In addition, presented contours (4th column) shows the velocity component along the height of the boiler (solid y-velocity component), with the velocity restricted to a maximum value of 0. In this way, it can be clearly noticed for the HELSCM model, at the walls of the computational domain, the velocity is less than zero, meaning that the particles in the core rise and at the walls gravitate down. Such behaviour is entirely natural within circulating and fluidized bed boilers. A layer of particles forms a film on the boiler's inner walls. These particles move down the walls towards the boiler's bottom and are subsequently lifted, primarily in the central region of the boiler. [130].

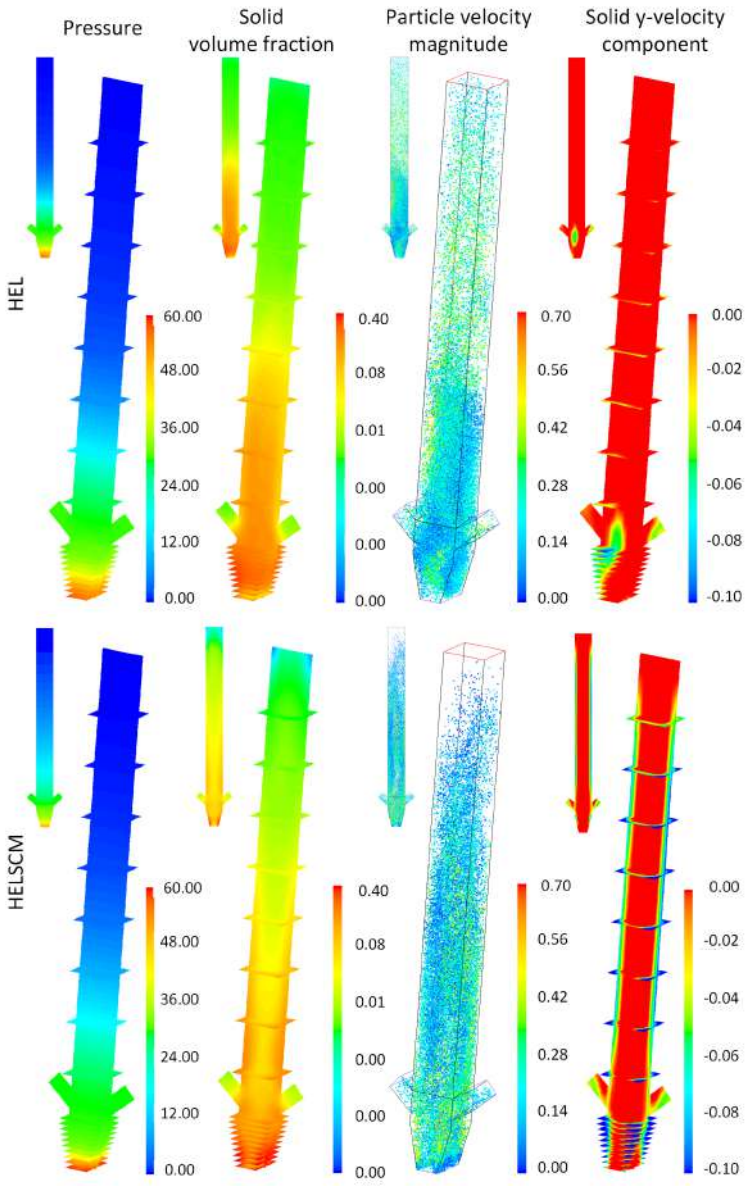

Fig. 5.25. Summary of simulation results in 10 s. The first row presents the results, derived from the HEL model. The second row, on the other hand, shows results from the HELSCM model. For comparison, the contours of the averaged pressure (1st column), the volume contribution of the solid phase (2nd column), the distribution of the particles in the computational domain, colored by their velocity (3rd column), and the solid y-velocity component (4th column) are presented. In addition, the planes are marked, from which the values for quantitative analysis were read

To verify the velocity of the solids, not only visually but also quantitatively, a lines in the cross-sections were created at four heights above the inlets (0.0215 m, 0.04875 m, 0.06825 m, 0.091625 m). For the comparison solid y-velocity component of the particle velocities was compared. Graphs confirm the visual results, i.e. the particle velocities for the HELSCM model are higher in the core and negative at the walls. In the case of the HELSCM model, the velocities are more uniform.

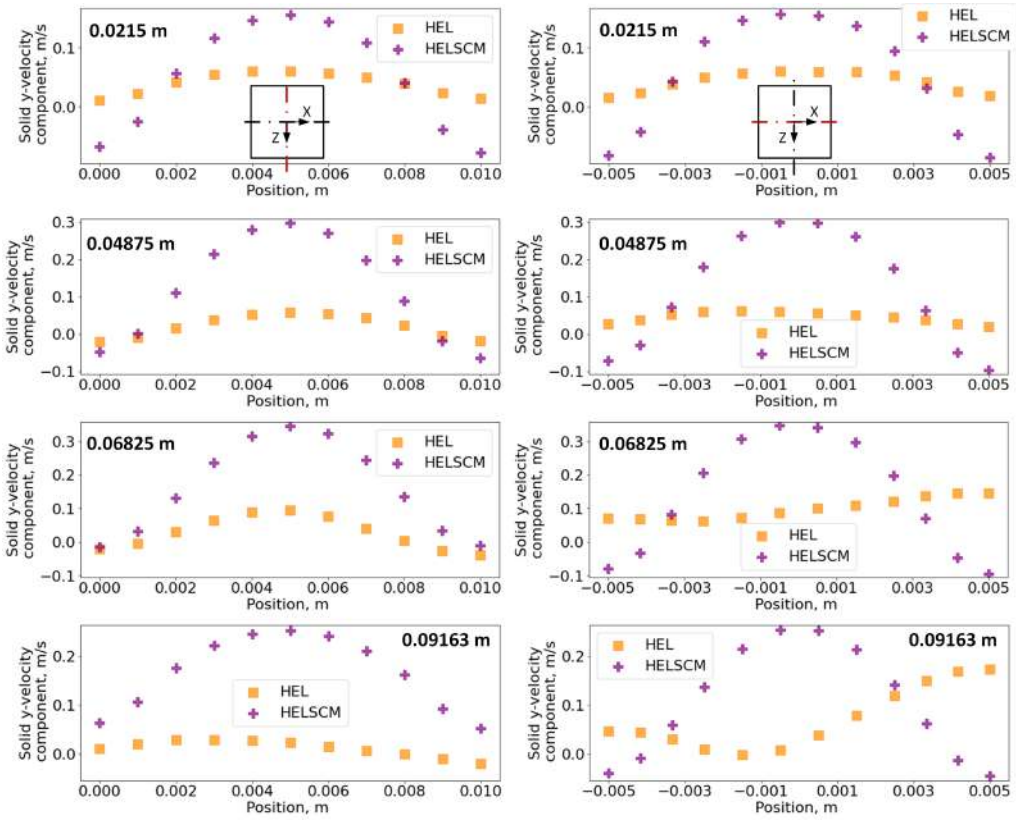

Fig. 5.26. Time-average profiles of the vertical component of the solid phase calculated at four horizontal cross section located at elevations, in two directions x and z determined in post-processing mode using numerical results obtained from HEL and HELSCM approaches

An additional comparison that was made was to measure the pressure drop along the height of the boiler. The average pressures, determined on the corresponding 18 planes, vary slightly. At the bottom of the boiler, the pressure is highest, and decreases with height. The discrepancies in values between the two models, are smallest at the lowest

parts, where there is the greatest accumulation of particles. Nevertheless, the differences are not significant. In the upper parts, the pressures coincide.

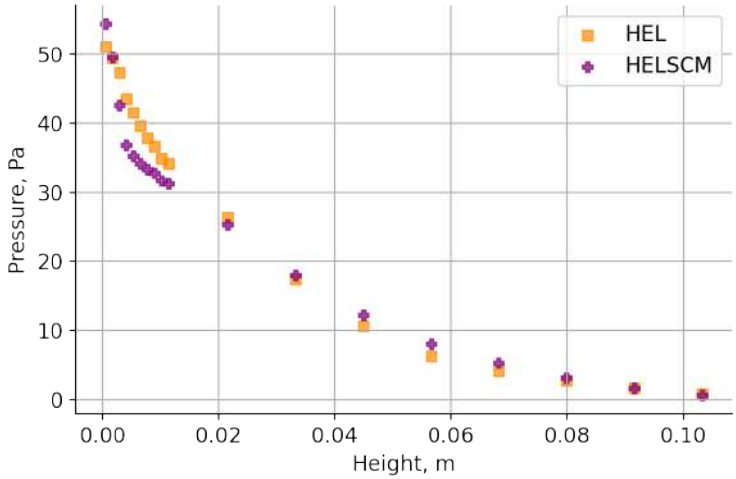

Fig. 5.27. Comparison of pressure drop between HEL and HELSCM

6
Hybrid approach for modeling large scale CFD boiler

The following section is focused on development of hybrid modeling approach for large scale CFB unit that will allows fast and robust calculation of the 3D model. This work was done, as the part of the project OPTI_AI_UNIT [1], being a cooperation mainly between Sumitomo SHI FW and Silesian University of Technology. The project concerns the development of a prototype, modular system based on artificial intelligence algorithms for increasing the reliability of the business. This was achieved by creating a hybrid infrastructure, including the optimization algorithms together with the adaptive predictive and prescriptive models working with control systems supported by the numerical models of selected critical equipment. This was consolidated using databases and reduced order models incorporated with algorithms monitoring the health of the power devices. The tested unit is 460 MWe power Łagisza Power block in Będzin, included in the list of International Energy Agency (IEA) reference facilities, as a reference object. The preparation of the module, and its testing, is carried out on the energy facility, but the use of the available techniques developed in this way can be applied to other types of facilities to improve their operation and reduce costs.

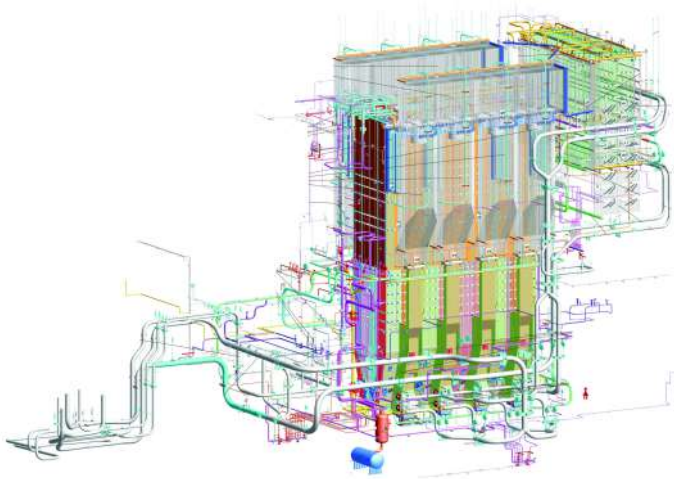

Fig. 6.1. Overall scheme of the circulating fluidized bed boiler, before simplifications made to the geometry used in the numerical simulations. A human figure has been marked in red, to show the difference in height, compared to the boiler.

The paper will not cover all the aspects of the complex project. It will primarily concentrate on two main elements: the development of models, and the exchange of data between fluidized bed boiler models, to enhance the overall system performance.

The components of the advanced intelligent control system, using machine learning algorithms, are the numerical models, developed based on the power unit example (i.e. Łagisza Power block). Those models are the digital equivalent of the analyzed device. The main idea is the development of numerical models, explained more clearly in next sections but listed below

- MEBal (Mass and Energy Balance) - mass and energy balance model of the boiler,
- SFW1D - one-dimensional model of the boiler,
- SFW3D - three-dimensional Euler-Euler model of the CFB boiler,
- EL3D - three-dimensional Eulerian-Lagrangian CFD model developed in ANSYS® Fluent Software,
- ELROM - the artificial neural network model.

To integrate the developed numerical models with the predictive-prescriptive algorithm, an algorithm will be proposed hybrid algorithm that allows synchronization of numerical models and correct exchange of data between them. The boiled one-dimensional zone model, hereinafter referred to as the SFW1D model, is developed to perform numerical simulations almost in real time. The calculation results are the additional information (input data) to the query coming from the control algorithm. The SW1D model, is intended to simulate the heat exchange, the composition of the bed (coke, ash, fractions of chemical compounds) and grain distribution. Despite many simplifications, SW1D model is crucial for protection of chaotic and incorrect decision affecting the stability of the operation of the power unit. To ensure it, an appropriate initialization of numerical calculations is required. For such purpose, the necessary data, suitably formatted and tailored to meet the requirements of the calculation code, will be sourced from historical data repositories. These repositories will tap into the comprehensive database of energy facilities managed by the Industry Partner Sumitomo SHI FW Energia Polska Ltd. If necessary, data might also be extracted from the repository of the steady state SFW3D.

Due to the number of cases analyzed, there might be situation, where SFW1D model is not sufficient to analyze the operating condition of the boiler. For this purpose, the steady state, three-dimensional SFW3D model was developed, to simulate the process of heat and mass exchange between phases. The model takes into account the combustion process, SO_2 and NO_x formation and reduction, material recirculation, heat exchanges within external heat exchangers, and various other physical phenomena. The main part of the model is a three-dimensional combustion chamber. Material recirculation is possible with the help

of sub-linked empirical models, coupled to the calculation procedure. Despite model's increased accuracy, and steady-state calculations, the extensive duration necessary for a complete simulation remains a significant drawback. It is a common feature for three-dimensional models, mentioned in this work. Therefore, those models cannot be directly implemented, as the component of the boiler control system.

Due to the limitations of the SFW3D model, as well as the stability of numerical calculations, it is necessary to support the operation of the model, the calculation algorithm requires the definition of a correct and real starting point. The set of such variables that are used to initialize the model are the velocity vector components for the gas phase, the velocity vector components for the solid phase, the volume fraction of the solid phase, pressures, and solid diameter.

To provide the required parameters to initialize the calculation, the Hybrid Euler-Lagrange model is used. In this chapter, this model is referred to the abbreviation EL3D. EL3D is the numerical model of CFB, developed in Ansys® FLUENT software. The proposed approach, allows direct tracking of the particles trajectories and the hydrodynamics of the bed. The complete and detailed model of the boiler includes the combustion chamber, steam superheaters along with other elements affecting the hydrodynamics of the bed. Due to the dimensions of the CFB boiler (27x10x48 m) and the variation in the scales of the spatial and temporal scales present in the various zones of the CFB boiler, the geometric model will be limited only to the combustion chamber. To maintain the material recirculation with including the heat exchange, an external mathematical model is utilized, and implemented using set of User Defined Functions. Within the mathematical model, the crucial aspect is to define correctly the boundary conditions, for boilers evaporators. For that purpose, these conditions are determined using AMIGO software, which has been developed for 60 years by consortium member Sumitomo SHI FW Energia Polska Ltd.

As with the SFW3D model, the EL3D model cannot be directly used to regulate boiler operation, primarily due to calculation time. Nevertheless, performing calculations using it improves the stability of the SFW3D model, as well as the accuracy of the results, when initializing the SFW3D model with outputs from EL3D. Therefore, it is not desirable to bypass this step. Struggling with long computation times, but unable to abandon simulations using the EL3D model, the solution lies in adoption of neural networks. The use of neural networks will allow the construction of a reduced model, that can return data with sufficiently high accuracy, which will be used as an initialization set for the SFW3D model.

The aforementioned models operate autonomously from one another. To integrate them, a repository of a results, as well as input data was established for each analyzed operating scenario of the boiler. In addition to the repository itself, it is important to

select the right tools to ensure that data is exchanged between the various models. It guarantees error-free data exchange between models. An approach of data exchange, will be covered in following subsections.

The subsequent chapter aims to describe the primary assumption of data exchange among all the models and development reduced order model of the CFB boiler by application of the Machine Learning algorithms. The main focus will be given on the numerical models and simplified models development to support optimization models.

6.1. Data exchange between individual models and system components

In connection with the use of simulation results from different models, executed in different software, different programming languages, it is necessary to create a space, which allows information to be stored and exchanged in a structured and easy way. A database (DB) is a structured collection of data, allowing for efficient storage, retrieval, and manipulation of the data. It is designed to manage large amounts of information and allows users to access and update information in an easy way. Databases are used in a wide variety of applications, including business, science, education, healthcare, etc. They can be stored on a single computer or distributed across multiple computers in a network. In the boiler module, five models were developed and applied for running hybrid approach:

To facilitate the utilization of simulation results from diverse models executed in different software and programming languages, it becomes imperative to establish a platform that enables structured and seamless storage and exchange of information. A database (DB) serves as a systematic collection of data, enabling efficient storage, retrieval, and manipulation of information. Its purpose is to handle extensive amounts of data and provide users with easy access and updating capabilities. Databases find application in various domains such as business, science, education, healthcare, among others. They can be stored on a single computer or distributed across multiple computers within a network. Within the context of the boiler module, a total of five models were created and implemented as part of the hybrid approach.

Since each model operates independently of each other, in the described approach, the calculations for each model are made on a separate virtual machines (VM). It is determined also by several additional factors, such as different programming languages used by the models, different software as well as different operators and preparers.

As calculations using individual models take place on separate VM's, each of them has oto be configured for the relevant model. In the case of ELROM, there are two machines. One, where the Artificial Neural Network (ANN) is built and one, where the predictions

are performed. Therefore, in order not to introduce too many complications, the name of the model remains the same, but when referring to the machine where the model is built, the *build* designation is in brackets. On the other hand, when referring to the machine where the prediction is performed, there is the designation *predict* in brackets.

To provide connection between the data itself and a database, an Representational State Transfer Application Programming Interface (RESTful API) was utilized. API allows multiple software components to communicate with each other using a set of definitions and protocols. RESTful API breakes the full transition into smaller modules, which addresses an underlying part of transaction. To obtain resources certain specific commands have to be used. It has to be noted, since the REST calls are stateless, nothing can be saved or remembered from one transaction to another.

A very detailed description of the RESTful API is presented in the work [56]. It ensures the security of data transmission, which is incredibly important for a system that exchanges sensitive data - such as continuous data on boiler operating parameters. In such industrial systems, it is necessary to ensure high security of the exchanged data and to meet the requirements of the many IT standards and protocols. For each model, a specially dedicated master codes responsible for communicating with the API, were also created. It was necessary, since the models were created using different programming languages and software, which prevents direct connection to DB. Additionally, the master codes, are not only responsible for connection, but also it is important to establish protective measures, in the event of failed connections or when attempting to call or return models with erroneous data.

In the Figure 6.2A, a diagram of the data exchange between the different elements is presented. Figure 6.2B, presents the exchange of data between the different models, this is of course done using the API. With this visualization, the path of data exchange between the different models is presented. Five models are used to provide a response characterizing the operational state of the boiler to the external prediction and prescription system. The execution time of the models is different, ranging from a couple of minutes to approximately 23 hours. First the data, from the system are provided the MEBal model, where with the important information (about the temperatures, pressures at specified points and some mass flows) the user is able to obtain the rest of required mass flows, needed as the input values to the SFW1D model. SFW1D is a zonal model, allowing numerical simulations to be performed, almost in real time. It can be used to simulate the heat transfer process in a bed, identify its composition including fractions of coke, ash, and chemical compounds, as well as determine the distribution of its grains. Although the model is highly simplified, it can help prevent control algorithms from making incorrect decisions that may cause instability in the operation of a power unit. The output data from SW1D is then used by SFW3D, but it is just a part of the input data. SFW3D, for

more precise result requires also additional data. Here, it comes from EL3D (obtaining results takes approx. 23 hours), or ELROM (predict) (prediction process with all the pre- and post- processing of data takes around 1 minute). The decision which results to use, is made on the basis of the input values. A more detailed description of the selection, is presented and described in the sections below, together with a more detailed description of the SFW3D, EL3D and ELROM model. Nevertheless, the data from SFW1D and ELROM are the initialization data for the SFW3D model calculations, allowing heat flow in the fluidised bed boiler to be modeled. The results obtained, are passed on to the prediction and prescription models.

Great emphasis was put on ensuring, the boiler subsystem is running quickly and efficiently at each stage. As a result, all subsequent models, as well as the API and database, were optimized to ensure fast execution.

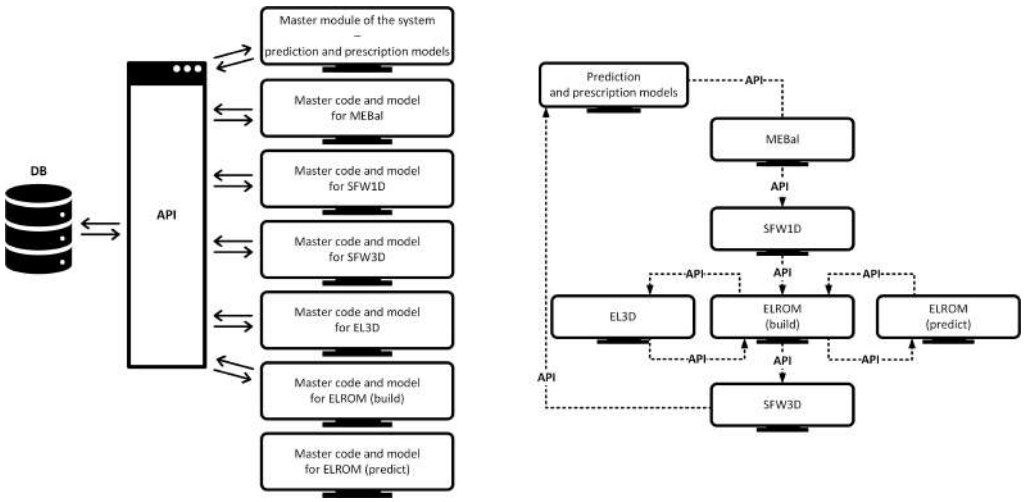

Fig. 6.2. **A:** Communication within boiler module - the IT infrastructure. **B:** Flow of data in the boiler subsystem

6.2. MEBal process model

MEBal is a calculation tool, developed to calculate the energy and mass balances of the boiler, over the specified control boundaries (Figure 6.3). The calculation relies on the principles of mass and energy conservation, using the indirect approach outlined in the established DIN 1942 standard [52]. The MEBal tool incorporates additional calculation procedures for determining the heat requirements of individual heat exchangers, fuel moisture and heat value, airflow rates, and fuel heat input. The developed tool returns the information of the boiler efficiency. In addition, the calculation outputs include data

on fuel and air flows, fuel heating value and moisture. These data are used as input values for the SFW1D model.

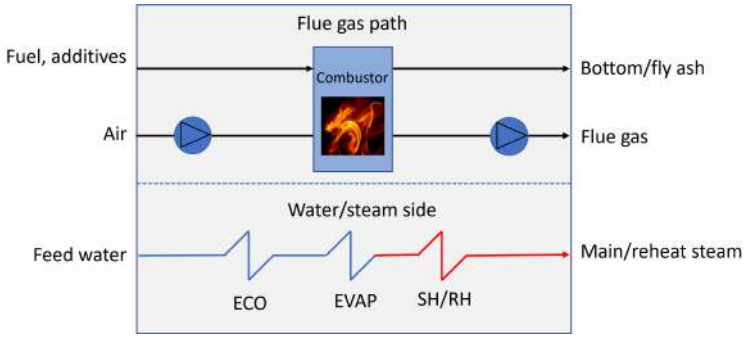

Fig. 6.3. Boundary Conditions for the MEBal tool

6.3. SFW1D process model

Another calculation tool, developed for the prediction of the CFB process performance is the SFW1D model. The model solves mass and energy balances for gas and solids. The system includes incorporated components specifically designed for the CFB hot loop, which encompasses the furnace, internal heat exchangers, solid separators, and external heat exchangers. The model's individual components, such as a furnace, were partitioned into specific calculation units vertically. The model incorporates chemical reactions such as combustion reactions of char and volatiles, as well as reactions involving limestone components. All the balance equations are solved through iterative processes.

The solids balance calculation is a crucial component of the model and is conducted in a 0D format for the entire CFB hot loop. The calculation involves determining the fractional masses of solid materials within the system, and calculating the mass flows of materials exiting the system. Each solid material is individually evaluated, and it is further categorized into six different particle-size fractions. The substances considered in the analysis consist of char derived from the fuel, fuel ash, make-up sand, and components of limestone ($CaCO_3$, CaO, $CaSO_4$). The primary factors influencing the solids balance are the particle sizes and comminution properties of the input materials, the effectiveness of the separator, and the operational conditions including fluidization velocity and bed pressure. To determine the vertical distributions of the solid materials, empirical correlations are utilized, although there is also an option to provide a solids density profile as input to the model, such as one based on the measured pressure profile.

The results from the SFW1D model calculation include the amounts of heat transferred to various components within the CFB hot loop as the rates of mass flow, the compositions

of the bed material, the particle-size distributions of the bed material, bottom ash, and fly ash. Additionally, the concentration profiles of solids and the flow of circulating materials through separators and INTREX units are also provided. The model's code is composed of numerous subroutines, each serving a specific purpose within the boiler process (see Figure 6.4). The predictive ability of the 1D model relies on empirically derived correlations, which have been validated through extensive measurements conducted in commercial-scale CFB units.

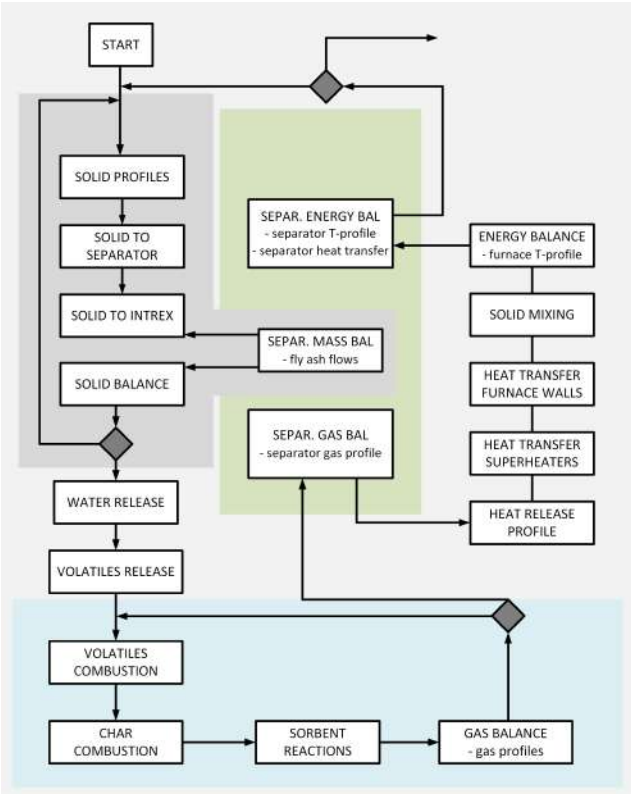

Fig. 6.4. Calculation flow sheet of the SFW1D model

6.4. Steady-sate 3D multiphase model

The main purpose of created numerical models is to support the development and design of CFB boilers, optimize processes, detect or analyze failure or problematic cases. This approach minimizes the operating costs of the real devices. In addition, modelling allows individual cases to be analyzed without taking the unit out of service.

Another of the mentioned models, important from the point of view of modeling and combining them into a whole, is the SFW3D model. It is a specially tailored version of the 3D multiphase general-purpose model, developed by LUT University [95]. The schematic working principle is presented in Figure 6.5.

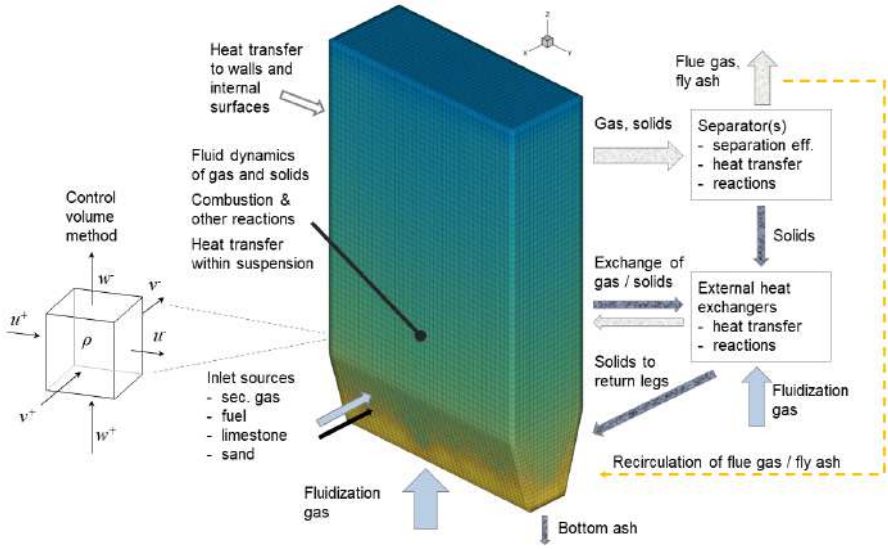

Fig. 6.5. Simplified geometry with marked boundary conditions

In the SFW3D model, the furnace is modeled in 3D. The other connected systems (i.e. separators, INTREX (superheaters) units, solids looping systems) are models by black-box models, exchanging data with the furnace model. The variable systems that have been resolved encompass various aspects such as the fluid dynamics of solid and gas, heterogeneus fuel reactions (such as evaporation, devolatilization, char combustion/gasification), reactions involving different types of sorbents (such as calcination, sulfation, and desulfation), reactions occuring uniformly throughout the system (such as combustion, shift conversion, and NOx reactions), as well as the transfer of heat through convection and radiation within the fluidized bed and between surfaces. The introduced approach utilizes a control volume method to address the diverse transport equations under steady-state conditions. The furnace domain is discretized by a structural body-fitted mesh with 55,440 calculation cells ($20 \times 40 \times 66$). The grid size was selected in order to strike an optimal balance between the accuracy of the calculations conducted and their computational time. The three-dimensional transport equations are resolved using a first-order upwind differencing scheme and the Gauss-Seidel method with successive overrelaxation. Alongside the fundamental transport equations for mass, energy, and

momentum, the model incorporates empirical correlations to establish various governing closure models. These closure models, such as those for solid concentration profiles and heat transfer coefficients, are crucial components of the model. The input data is supplied as a text file through the API, encompassing variable parameters such as feed rates and compositions. On the other hand, the fixed model parameters like geometry and mesh are hardcoded. The solver code is implemented as a dynamic link library and is invoked by the master code. The solver operates in two distinct modes: within the SFW3D code, it resolves the fluid dynamics of gas and solids with simplified flow fields, or it employs flow fields generated by EL3D or ELROM for more intricate and detailed flow simulations.

Convergence is assessed based on several criteria, including the errors in mass and energy balance (limited to less than 0.1%) and the absolute difference between consecutive iterations of monitored parameters such as bed temperature, exit temperature, and exit gas composition. If the code is called without externally determined flow fields, the calculation time ranges from 4 to 7 minutes. However, if external flow fields are determined, the calculation time increases to 5 to 9 minutes. The reason is due to the more complex flow field, influencing the model's convergence. Once calculations are done, the model saves the overall results of the simulation, as well as cross-sectionally averaged 1D profiles of various variables, in the form of text files. The 3D process data is stored in a binary file for visualization purposes in ANSYS® Fluent.

For more details, please refer to earlier literature, where the theory behind the operation of the model is described in detail [95, 99, 100, 98].

6.5. Geometry and mesh of the CFB boiler - EL3D

The presented geometry was created on the basis of an real scale boiler, located in Będzin (Poland), which is part of the Łagisza power plant. The boiler has a capacity of 460 MW, the rectangular cross-section of the combustion chamber has dimensions of 10.6 x 27.6 m. The height of the combustion chamber is 47 m. The height of the cooling tower is equal to 133.2 m [147]. In the real boiler, there are also 8 cyclones attached to the combustion chamber, which role is to separate the flue gases from the particles and are responsible for returning not burnt material, or inert material, back to the combustion chamber. The boiler is symmetrical, which means that its left and right sides are the same (the number of primary/secondary inlets, oxydixer inlets, recirculated material inlets and fuel inlets are equal).

In order not to introduce too many complications to the numerical calculations, the geometry was simplified by the elimination of the 8 separators and external superheaters. However, to take into account their presence in real device, so to ensure recirculation of the material, a UDF was used, the exact operating principle is described in 5.6.1. The

recurirculated material is returned through the intrex recurculation inlets (visualized in Figure 6.6). The difference is the injection type. Instead of returning particles do specified cells in volume, the material was injected through the surface.

The numerical mesh and geometry were created using Ansys® Workbench. The used mesh, in its final version, was created based on previous experience with the HEL approach. A detailed description, summarizing the experience on grid selection and geometry is in [55, 12, 10, 77]. A crucial condition for the numerical cells was to select the suitable mesh resolution in order to prevent the solution from reaching the solid-packing limit, which is utilized by the kinetic theory of granular flow (KTGF) [77] to simulate interactions among solid particles. This matter was thoroughly explained in reference [11]. The numerical mesh, used for calculations, consists of 816,407 elements, with an average element size of 0.252 m. The shared topology utility was employed to create continuous and seamless grids with conformal properties. As the result, the following quality statistics were obtained: element quality – average: 0.75 (min: 0.23 – max: 1.00), skewness – average: 0.05 (min: 0.0 – max: 0.73), aspect ratio – average: 2.15 (min: 1.00 – max: 6.60).

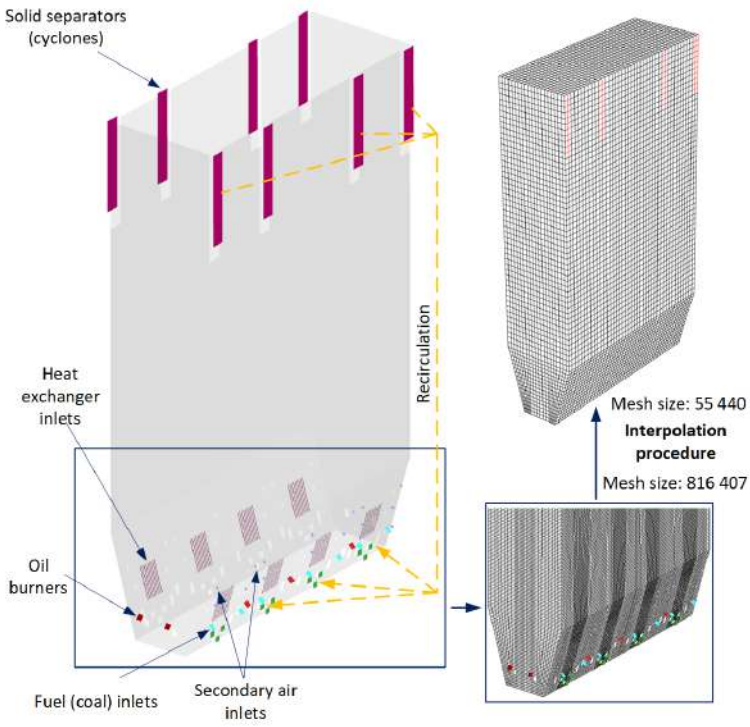

Fig. 6.6. Simplified geometry with marked boundary conditions

The time needed for time averaged solution for the *cold* simulations (i.e. without considering species transport and heat transfer), was approx. 23 h. The presented simulations were carried using 1 PC, equipped with 48 cores, 96 logical cores and 96 GB operating memory. The set of closure models and parameters used in the numerical simulations for are summarized in the table 6.3. Due to the simplifications in geometry used, appropriate boundary conditions (see tab. 6.2), were assigned to individual surfaces in the boiler to ensure conditions as close as possible to the operation of the complete unit.

Table 6.1

Closure models and parameters used in the simulations, where model names are given as they appear in ANSYS®Fluent CFD code

Solid pressure, p_s	lun-et-al [85]
Radial distribution, $g_{0,ss}$	lun-et-al [103]
Granular viscosity, μ_s	gidaspow [58]
Angle of internal friction	30°
Granular bulk viscosity, λ_s	lun-et-al [85]
Frictional viscosity, μ_{fric}	schaeffer [116]
Frictional pressure, p_{fric}	based-ktgf [124]
Radial distribution	lun [85]
Granular temperature, Θ	algebraic [124]
Drag model, K_{sg}	gidaspow [58]
Frictional modulus	derived [146]
Elasticity modulus	derived [146]
Restitution coeff. in solid phase, e_{ss}	0.9
Friction packing limit, $\varepsilon_s^{\text{fric}}$	0.61
Packing limit, $\varepsilon_s^{\text{pk}}$	0.63
Transition factor of HEL	1.0
Normal discrete phase reflection coefficient	0.8
Normal discrete phase reflection coefficient	0.5

Table 6.2

EL3D boundary conditions

Boundary	Number of inlets	Boundary condition
Primary air	1	mass flow inlet
Secondary air	32	mass flow inlet
Oil burners	7	mass flow inlet
Fuel inlets	14	mass flow inlet for gas phase flow rate (by injection) for solid phase
Recirculation inlets	24	mass flow inlet for gas phase UDF for solid phase
Heat exchanger inlets	8 (multi surface)	mass flow inlet

6.6. Field values determined using neural network models

To speed up the process of obtaining results, for the new boiler load conditions, it was decided to use artificial neural network (ANN). Such solution, constantly growing in popularity, offers high accuracy of results, obtained in a short time. Deep learning regression [20] models, make it possible to achieve predicted results, in a several seconds, which is considerably less time than a 23 hour full CFD calculation. The field values to be calculated are 55 440 (resulting from 66 cells in height and 840 forming a single horizontal plane in the boiler). The field values that must be returned, corresponding to 9 Fields, which are then saved in repository, are listed below

Table 6.3

Fields interpolated between models

Field type	Field no.
x velocity component of solid phase based on favre averaging	Field #1
y velocity component of solid phase based on favre averaging	Field #2
z velocity component of solid phase based on favre averaging	Field #3
x velocity component of fluid phase based on favre averaging	Field #4
y velocity component of fluid phase based on favre averaging	Field #5
z velocity component of fluid phase based on favre averaging	Field #6
pressure averaged in time	Field #7
solid diameter averaged in space and time	Field #8
volume fraction covered by solid phase averaged in time	Field #9

It is important to map out the approach to solve the problem of predicting the field values, i.e. what data to take as an input and as an output. One of the ideas, which is very intuitive, was to take as input data the initial boundary conditions of the CFB boiler and the position of a cells. In such way, for one case there would be 55,440 entries, each with 28 features (26 corresponding to the BC's and 2 corresponding to the position of the cell). The expected total number of calculated cases reached 254. Therefore, using the approach described above, there would ultimately be over 13 million entries, each having 28 features. To reduce the amount of data entered into the model, another idea was tested and then used in this work. It involves dividing the computational domain into planes. Each plane, at the appropriate height, is given its own number (from $0 - 65$). At each plane there is 840 cells. As a result, the number of records entered into the each model is reduced to approximately 16,000. However, the prediction of 840 values at once might decrease accuracy. The values in each cell can differ significantly from each other, as a result, i.a. of the location in the computational domain (e.g. the Z-component of the velocity along the height), where the gas/solid phase velocities are much higher in the boiler core than at the walls. Thus the cells with similar value prediction errors were grouped, to be the outputs to the sub-models.

To determine prediction errors for single cell and to be able to decide to which sub-models assign individual cells, time- and computationally expensive models were made. The data from 249 cases was divided into three subsets: train, valid, test (70%, 20%, 10% respectively). In such way, the input and output data for building single models was prepared. For each model the hyperparameters used during training were set to default, there was no optimization applied. For each cell, the model was built five times. After each build, the MSE value for the test set was determined and it was saved to a file. Afterwards, the process was repeated for the next cell. The above algorithm is illustrated in Figure 6.7 presented on next page.

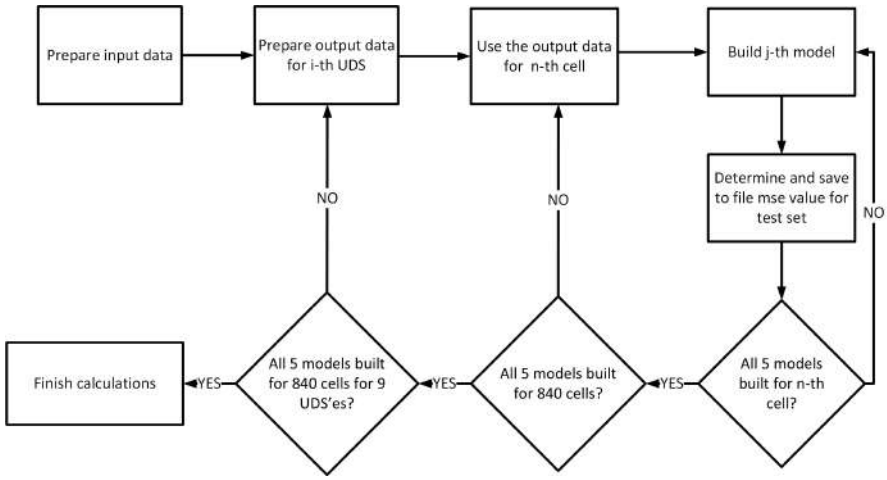

Fig. 6.7. Algorithm describing the process of building predictive models for single cell

Once the data generation and collection process was complete, an additional script was written, which had the task of finalizing the assignment of specific columns to the individual sub-models for a given Field. The allocation criterion, to specific sub-model, was determined by the error value, and are presented in the table 6.4 below

Table 6.4

Error ranges determining the submodel group

ELROM submodel no	Error range
submodel 1	<0-0.1)
submodel 2	<0.1-0.2)
submodel 3	<0.2-0.3)
submodel 4	<0.3-0.5)
submodel 5	0.5≤

After reading the file containing the data with the error values, the average error of the five determined was calculated. On this basis, the corresponding cell number was assigned to the target group. The process was repeated, until all cells were grouped, for a given scalar. The illustrative Figure 6.8, shows the cells from each subgroup as well as the concept itself of predicting values for a given plane. Please note that it is only a visualization of the idea. For each of the subgroups, the distribution of the cell groups, on the plane, may differ significantly from each other.

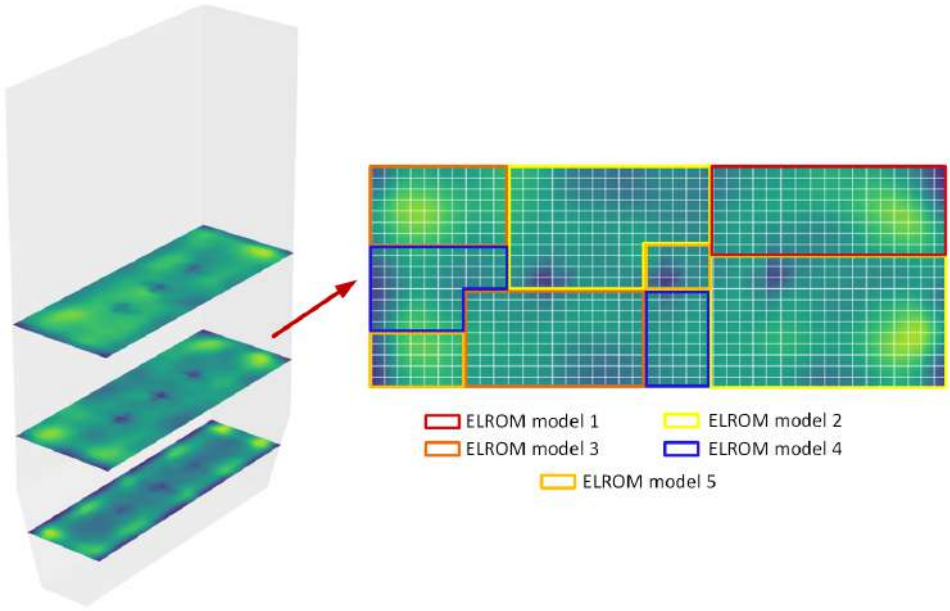

Fig. 6.8. Visualization of the division of the computational domain into slices, and the cells making up the slices into appropriate subgroups

Once the appropriate cells were assigned to each subgroup, the construction of the actual models began. At this stage, when the final number of models is lower, and also the priority is not to determine the approximate prediction error, but the highest possible accuracy, each model construction is preceded by a search in the space of hyperparameters, to provide such accuracy. It is an important step that must be carried out for each sub-model. Due to the different ranges of output values, as well as the different number of outputs, each model architecture will differ. The hyperparameters, changed during optimization were:

- learning_rate - determines the size of the adjustments made to weights at each iteration of the algorithm
- n_layers - layers in between input and output layer, where neurons take in a set of weighted inputs to give an output through an activation function
- n_units - low level features for every hidden layer
- batch_size - the total number of train samples in single batch

During the search for the best hyperparameters set, as well as at the stage of building final models, some parameters were unchanged. For all configurations, the Adam estimator was set, which is a relatively new estimator [75, 113, 53]. It uses first-order, gradient-based

optimization of the stochastic objective functions, based on adaptive estimation of lower-order moments. It is a recommended estimator for machine learning problems with large datasets and many parameters [75]. Also an early stopping with restoring the best weights was applied. It means that, if the result does not improve (the accuracy score does not change), in subsequent steps the model-building process stops, and the best weights are restored and applied to the final model. As the metrics, to determine the accuracy of the model, mean squared error (MSE) was chosen.

To recall, the main idea is to create a group of models with which it will be possible to predict field values, for a specific plane. The process of building models begins with the preparation of the data, and dividing it into three appropriate subsets (train, valid, test). Then the standardization technique is applied [89], which influences and leads to a stable solution and high precision in predicting target parameters. It rescales the data relative to their mean and standard deviation, according to formula [93]

$$\mathbf{X}_{\text{sc}} = \frac{\mathbf{X} - \mu_{\text{train}}}{\sigma_{\text{train}}} \tag{6.1}$$

where $\mathbf{X}_{\text{sc}}$ is the standardized value, $\mathbf{X}$ is the original value, μ_{train} and σ_{train}, respectively, denote the mean and standard deviation determined for the training set. Then, using the calculated μ_{train} and σ_{train}, the test and validation sets were standardized. The space was then searched to find the best hyperparameters set. Once the best one was determined, it was saved and then, five similar models were built. Each time, after the building process, the MSE metrics were saved. Having five models, the final one from them, with the highest accuracy was selected, and it was used for prediction purposes. By repeating the building process five times, the stability of the model was tested, and it allowed to select the best possible version. The whole process, was performed for each sub-model, for specific field. Prediction, like model building, must be preceded by standardization of the input data. Then, for a given FV subgroup, the corresponding model is called up, which returns a prediction of the value, based on the input data. The methodology for both, model building and prediction for each field is similar, and presented in Figure 6.9 (left and right respectively).

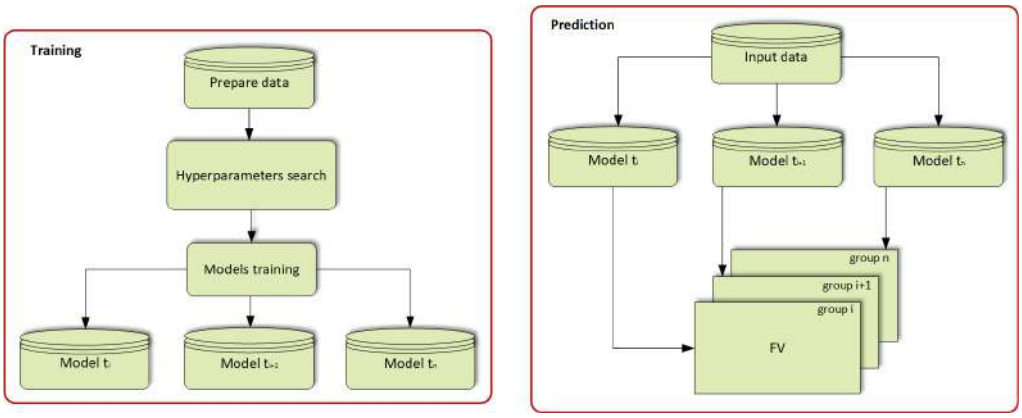

Fig. 6.9. **Left:** Training process, preceded by a search for hyperparameters set, depending on field final number of models is from two to five (n). **Right:** Scheme pf prediction procedure, to obtain all values for particular layer

6.6.1. SFW3D and ELROM models results

During a data connection and preparation, five cases were excluded from all available recalculated cases at the simulation stage to determine the accuracy of the prediction. In the following, results for only two cases (64% and 98% of boiler load respectively) are presented. An important factor, when building predictors using ANNs, is the amount of data used for training. The greater the amount, the better the accuracy of the model. Therefore, the comparison will be made between the two configurations. The first (ELROM v1), for which 83 cases were used for training, and the second (ELROM v2), for which 249 cases were used.

A quantitative comparison, based on the mean square error, is presented on the next page. The prediction errors, for two fields (Field #2 and Field #8), are juxtaposed. Both for 64% and 98% boiler load. In the case of Field #2, and boiler load equals to 64% the errors, are similar for both model variants. Significantly greater difference, and a reduction in error for the v2 variant, can be seen for the volume fraction of the solid phase (Field #8). The similar observation is made for a boiler load of 98%, where the MSE drops significantly for submodels 1 and 2, variant V2. The differences in prediction errors are similar, for Field #2, for boiler load 98%. There is no significant improvement, for variant v2 with respect to v1. Additionally, further at the qualitative comparison stage, the differences are not significantly noticeable.

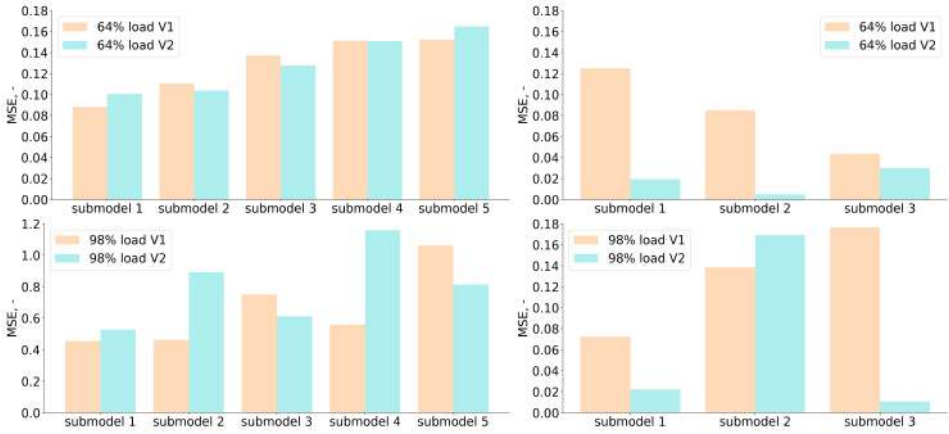

Fig. 6.10. Comparison of the MSE for prediction: **left** Field #2 (z - component of solid velocity), **right** Field #8 (Volume of Fraction covered by solid phase averaged in time). The top row indicates to the 64% load, and bottom row to 98% load. The model variant ELROM v1 is marked with orange, and the variant ELROM v2 is marked with the blueish color

One of the most important values when it comes to particles moving in a fluidized bed boiler is their velocity, along the height of the boiler. In Figure 6.11, it can be noticed that the greatest solid velocity is in the in the middle of the boiler height, while on the bottom and at the top it decreases, for the boiler load equal to 64%. For the boiler load 98%, there are local vortices at the bottom of the boiler which increase the solid velocity. However, regardless of the use of the ELROM v1 or v2 model for particle velocity prediction, the results are a good representation of the velocities present in the original simulations using EL3D.

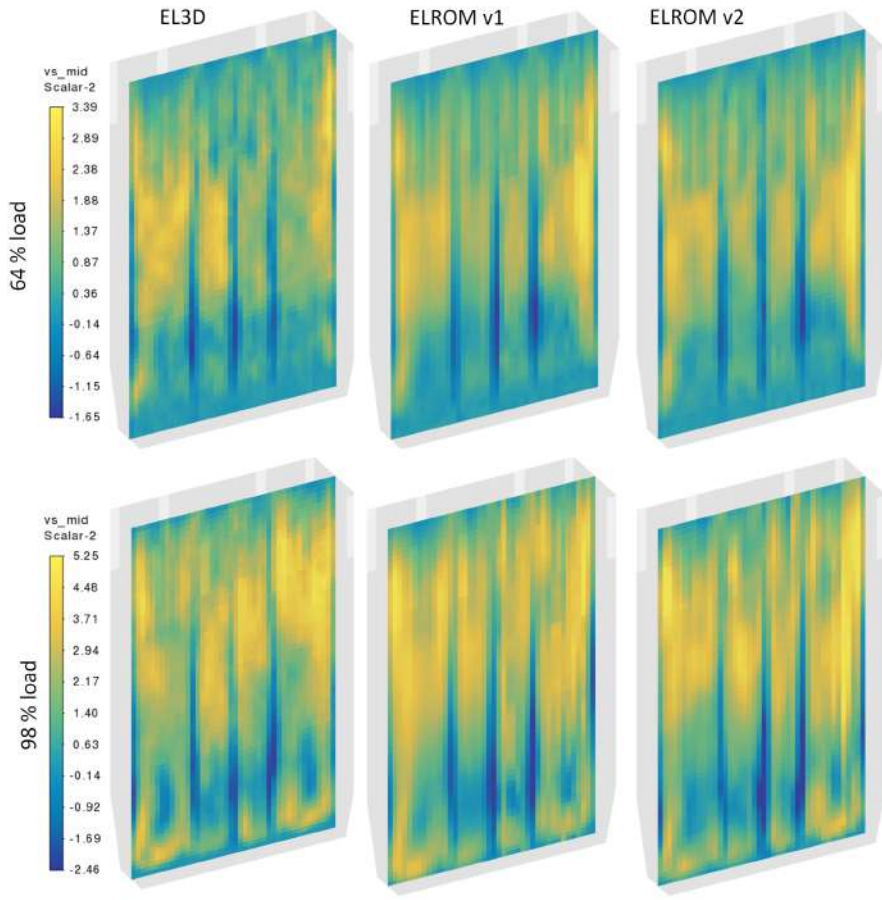

Fig. 6.11. Comparison of the Field #2 (z - component of solid velocity): model EL3D (left side), reduced model ELROM v1 (centre), reduced model ELROM v2 (right side). The upper row presents result for the 64% boiled load, while bottom for the 98% boiled load

The solid volume fraction holds significant importance in the field, as it is a crucial parameter characterized by a narrow range and the shape of the field. It reaches its maximum value at the lower region of the calculation domain, gradually decreasing to nearly 0 towards the upper region of the combustion chamber. This pattern is clearly observed in the results obtained from the original EL3D simulations, where the highest concentration of particles is found at the bottom of the boiler. As the height increases, the contribution of these particles to the overall volume diminishes. In the case of the ELROM v1 model, a similar variation in particle concentration with height is observed. However, the highest concentration is primarily concentrated in the lower layers, while it decreases in the higher layers. Nonetheless, the solid volume fraction remains relatively

uniform in the remaining parts of the field. The ELROM v2 model, which predicts the volume fraction, performs significantly better. Once again, the highest concentration is observed at the bottom but decreases as the height increases, reaching its lowest point in the uppermost regions. The qualitative results are presented and depicted in Figure 6.12.

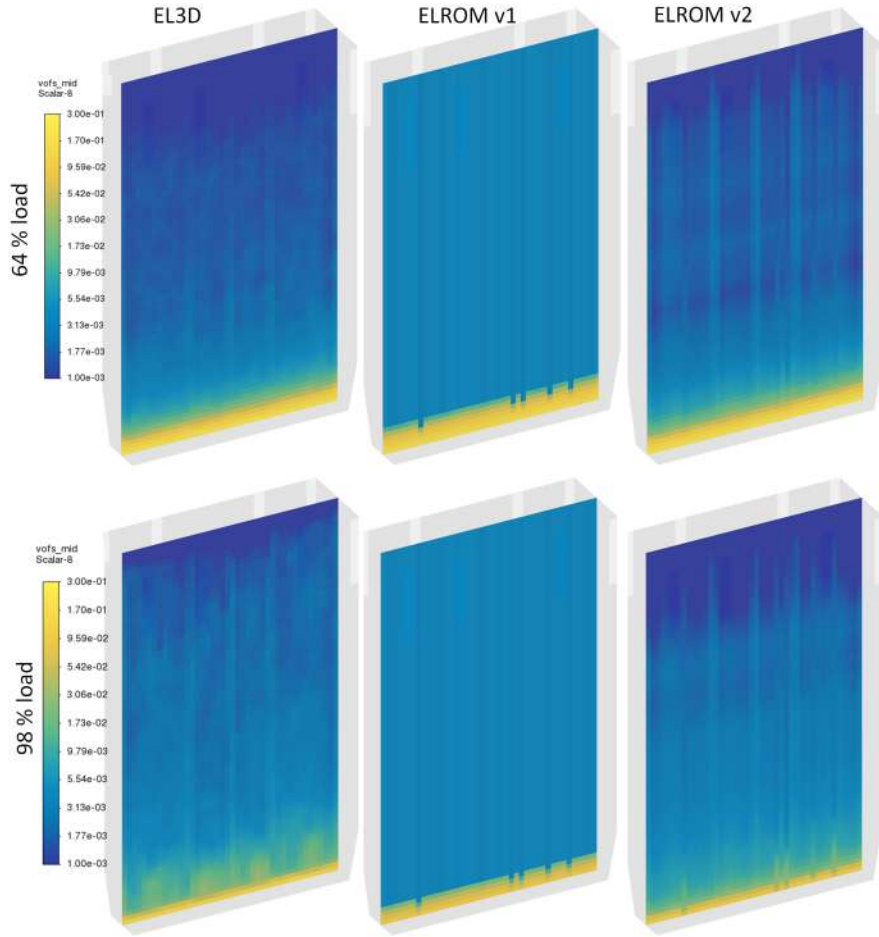

Fig. 6.12. Comparison of the Field #8 (volume fraction covered by solid phase averaged in time): model EL3D (left side), ELROM model v1 (centre), ELROM model v2 (right side). The upper row presents result for the 64% boiled load, while bottom for the 98 % boiled load

To compare the differences in the results of the SFW3D model, depending on the start of calculations with and without specific initialization points derived from the EL3D model, the results are presented at Figures 6.13 and 6.14 presents the flow fields affected by externally solved flow field (images a). With original solver the filed was quite uniform

in both cases. When the modifications were applied, i.e. the calculations were initialized with outputs from EL3D, the SFW3D followed the externally solved fields (images c). Without initialization using the results from the EL3D model, the results were more uniform across boiler sections (images b).

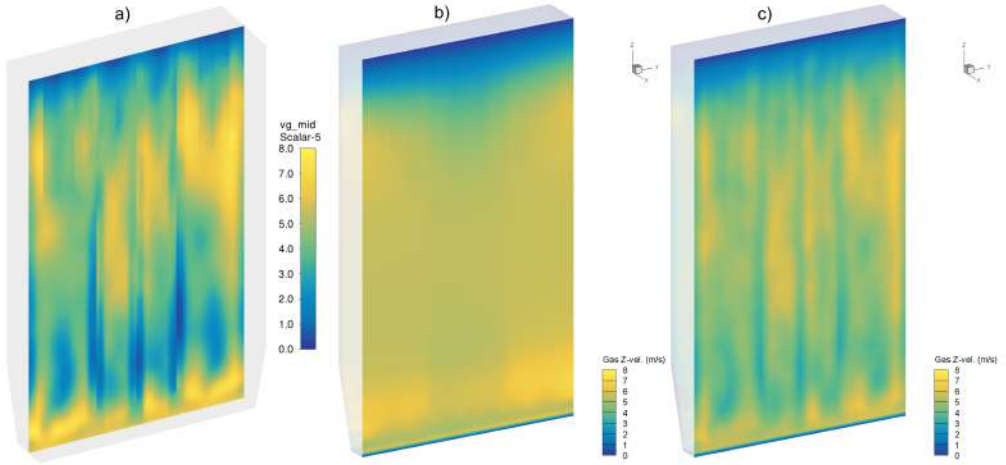

Fig. 6.13. Effect of externally solved gas flow field for the vertical gas velocity a) solved by CFD, b) SFW3D result without modification, c) SFW3D result with modification

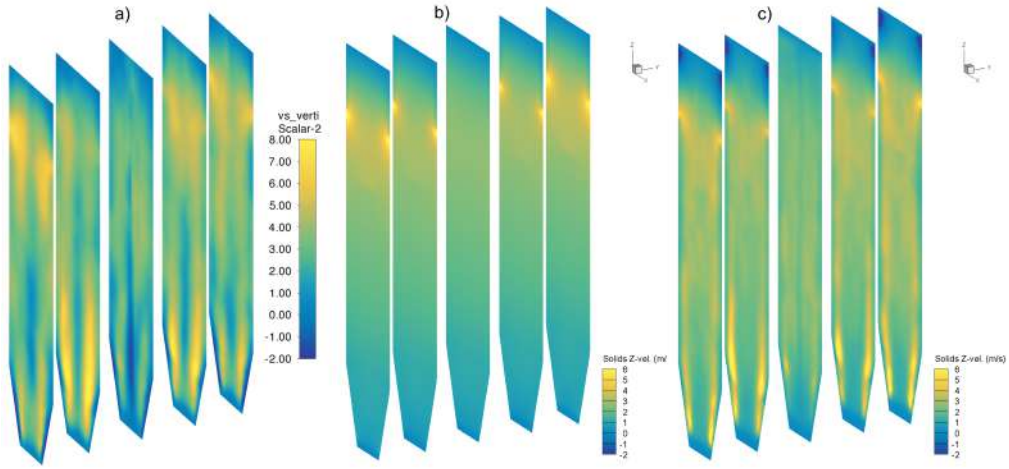

Fig. 6.14. Effect of the externally solved solid flow field for the vertical solid velocity a) solved by CFD, b) SFW3D result without modification, c) SFW3D result with modification

7
Summary and conclusions

In the presented dissertation, the main issue addressed is the use of computer fluid mechanics, to model granular flows. Granular flows are characterized by high solid volume fraction, where a very large number of particles, affects a significant number and complexity of collisions between them. The work mainly focuses on description of three already available approaches, used widely for modelling multiphase flows, i.e. Euler-Euler (EE) approach, Hybrid Euler-Lagrange (HEL) approach and Discrete Element Method (DEM). Of the three mentioned, the latter two were used to show the significant differences between them, mainly in simulation time as well as the approach to collision detection, and thus the behavior of the particles. Furthermore, machine learning algorithms are described, specifically concentrating on techniques used in presented work, i.e. decision trees and neural networks. Over past few years, machine learning has gained increasing attention and is progressively being integrated into various industries, often in combination with numerical computing approaches.

The work is essentially divided into two parts, where simulation results are compared with experimental measurements. The first, covering chapters 4 and 5, describes the approach of combining the HEL model and the DEM, which finally forms the Hybrid Euler-Lagrange Surrogate Collision Model (HELSCM) model. The main idea is to combine the advantages of the two models mentioned above. The HEL model, has proven effective in simulating multiphase flows within industrial fluidized bed boilers, primarily attributed to its relatively short computation times. However, it is not without disadvantages, including issues concerning calculation stability and collision detection itself, which is strongly dependent on the solid volume fraction in computational cell. On the other hand, the DEM model, despite its enhanced precision in collision detection, based on particles position, is not widely used in simulation of real-world objects of great size, primarily due to extended computational time. To present, in a more visible way for a better comparison, the differences in collision detection between the HEL model and DEM, a set of low-level tests was conducted. One test involved collisions between two streams of particles moving from opposite directions. The geometry consists of the main channel with two side inlets attached to it, through which particles are supplied. To ensure a valid comparison, identical boundary and initialization conditions were established for both models. In case of HEL model, the opposite streams at the time of collision, barely interacted with each other, due to the small volume fraction of the solids in the numerical

cell. In contrast, the DEM model correctly detected collisions. The particles, visibly interacted with each other in the center of the main channel, and then, as a result of the collision, changed their trajectory and moved along the center of the channel. In addition to the qualitative comparison, a quantitative comparison was made by determining the sections, in which the average particle velocities were determined. These values allowed to make their velocity profiles in the main channel. The aforementioned geometry was simplified, but its fundamental components were still mapped to a specially built test-rig. It features a transparent material of the main channel. With the test-rig, it was possible to make measurements reflecting simulations to have a reference point for the particles behaviour in reality. To ensure the repeatability of the measurements, they were made nine times. Measurements were captured by placing a high-speed recording camera in front of the test-rig. To determine the average particle velocity during the carried out experiments, in the relevant sections, it was necessary to create custom algorithm, developed in LabView software. The algorithm allowed to track particles based on the subsequent images captured during the measurements.

The subsequent step, after defining the benchmark results, was to create new approach for modelling multiphase flows. The main idea was to combine the strengths of the HEL and DEM, i.e. short simulation time and the collision detection precision. For this purpose the machine algorithms were used, which are characterized by the high accuracy and the speed of obtained results. Using the DEM model, a series of simulations of two-particle collisions were performed. Each time, the velocity components of both before and after the collision were recorded. With the data collected in this way, a simplified model, able to predict appropriate outputs based on the input data. In this case, the components of velocity vectors served as inputs and outputs. The model thus built was implemented for simulations using the HEL model, replacing the basic approach to collision detection, using User Defined Functions (UDF). The first test to validate the newly created model, named Hybrid Euler-Lagrange Surrogate Collision Model (HELSCM), was to make simulations of few particles moving from opposite directions, to check if the collisions occur. For HEL approach particles interpenetrated each other, while for DEM and HELSCM interacted, and changed the directions. Another test, reflecting experimental measurements, was performed, identical to simulations previously made with the HEL model and DEM, i.e. collisions of two opposite streams in a channel. In this case, also in addition to qualitative comparison, data was collected for quantitative comparison. For this purpose, average particle velocities in specific sections along the main channel were determined. Then, the results were gathered on a common graph in the form of velocity profiles, and compared with each other. In terms of qualitative comparison, the particle streams for the HEL model are interpenetrating, particle interactions are noticeable slightly. The DEM model shows the greatest agreement with experimental results, that is, after the particle streams

collide, they change their trajectory and move along the main channel. The HELSCM model, also shows greater detection of particle collisions than the HEL model, however, after collisions the particles are more dispersed in the channel.

The possibility of using the HELSCM model has also been tested in simulations of the circulating fluidized bed simulation phenomenon, in a boiler of small size. For this purpose, a simplified geometry of the fluidized bed boiler was created. Fluidization simulations were carried out for two models HEL and HELSCM. In order to compare the results with each other, results were collected from the volume shares of the solid phase in the domain, particle velocities, and pressure profiles were made along the height of the boiler. Particle velocities as well as volume solid fractions are similar. However, in the case of the HELSCM model, particle velocities near the walls are negative (they move downward), such phenomenon is desirable in fluidized bed boilers. The velocity profiles differ slightly near the bottom of the boiler, nevertheless, they become similar together with height.

The second part of the work focuses on a slightly different approach to modeling multiphase flows. In this case, as part of OPTI_AI_UNIT project, the goal was to map field values in a real-size boiler in Lagisza power block in Bedzin. To help monitor the help of power devices, as well as to prevent unplanned failures, several numerical models have been created. One of them is the 1D (SFW1D) model and 3D models: steady state (SFW3D) and the more complex EL3D (Hybrid Euler-Lagrange). SW1D model requires data from SFW3D model. The SW3D model returns more exact results, if is initialized with the data from EL3D model. The SFW3D model considers the process of the heat and mass exchange, while EL3D allows to track particles and the hydrodynamics of the bed. However, in case of 3D models the calculations are too expensive in terms of time, to be used for real-time boiler control. Therefore, neural networks are used as the basis for creating a simplified model, whereby results corresponding to those using the EL3D model are obtained much faster. The complete procedure for creating a set of models to reproduce the fields, for each of the required parameters, is described in detail. The set of those models was names ELROM. The data used to train the model ELROM, came from model EL3D. Based on the specified boiler operating parameters, they are able to return the corresponding field values. These values, regardless of whether they come from model EL3D or ELROM, are used to initialize the calculations of model SFW3D. The collected results clearly show that more the data for learning neural networks, the closer the obtained result is to the expected response. In addition, the initialization of the calculation of SFW3D models, with data from the EL3D model or ELROM model definitely increases its accuracy. It should be noted that the exchange of data between the mentioned models is not trivial, due to the different languages in which the programs are written and their different locations. Therefore, a special architecture for data storage and exchange using databases was developed.

7.0.1. Future research

The work described in the presented thesis is divided into two parts. The part on the construction of neural networks to predict value fields has essentially been done and completed. The next steps will be to deploy the developed models into production, to continuously help optimize the performance of the predictive-prescriptive system. This part, however, will be led inside Tauron Group.

The second part of the described work focused on development of the HELSCM model, as a combination of the HEL model and the DEM model. In case of that model there is still some room for improvement. According to that, the further work focus will be revolve around striving to closely match the DEM model, not only in terms of velocities - which at this moment are similar, but also in terms of particles trajectories themselves. It might require to create the data set or just to update existing data set. the new model will have to be built. In both cases, it will be necessary to update the existing model. If adequate accuracy or satisfactory particle behavior will not obtained, an attempt will be made to build a suitable dataset, based on other software used for particle collision modeling. Available, for this purpose, are software such as EDEM supported by ANSYS®Fluent, RockyDEM, or Altair® EDEM™. From an open source softwares, at least two are taken into account, in which modelling of particle interactions is possible, i.e. LIGGGHTS or OpenFOAM. Regardless of the alternative chosen, it will be necessary to gain knowledge and some experience. Again to verify, the accuracy of the improved models, the same experimental test-rig will be used, allowing to track the behavior of two colliding opposite streams of particles.

Simultaneously, the work will be carried out to develop approach and to create a data set to accurately reflect particle collisions of different diameters. In this case, also, the already used experimental test-rig will be used to obtain measurements results, which will be a benchmark case for simulations. The main assumption of the tests that will be carried out is particle collisions, where opposite streams will have different particle diameters. In parallel, work is underway to develop an approach to building a model and a set of suitable datasets to reflect particle collisions of different diameters. In this case, too, the experimental bench, whose measurement results were used in the target simulations, will be used as a reference. The primary premise behind the forthcoming tests is the examination of particle collisions in situations where opposing streams feature particles of differing diameters.

After the tests of fluidized bed simulation in a small sized boiler are done, the next step will be to improve the UDFs already written to simulate the fluidization phenomenon in a boiler of real size. In this particular case, the focus will be on trying to simulate the bed in a boiler geometry that mirrors the CFB boiler located at the Łagisza power plant, in

Będzin. The work will require modifications to the code used in the UDF, mainly involving parallelization. With it, it will be possible to simulate the several million particles involved in fluidization in a real-sized boiler.

Abbreviations

2D	two-dimensional	3D	three-dimensional
ANN	Artificial Neural Network	API	Application Programming Interface
BFB	Bubbling Fluidized Bed	CFB	Circulating Fluidized Bed
CFD	Computational Fluid Dynamics	DDPM	Dense Discrete Phase Method
DEM	Disctrete Element Method	UDF	User Defined Function
HEL	Hybrid Euler-Lagrange	EE	Euler-Euler
KTGF	Kinetic Theory of Granular Flow	MAE	Mean Absolute Error
ML	Machine Learning	MSE	Mean Squared Error
MP-PIC	Multiphase Particle in Cell	PC	Pulverized Coal
PFC	Particle Flow Code	PSD	Particle Size Distribution
RDF	Refuse Derived Fuel	REST	REpresentational State Transfer
RMSE	Root Mean Squared Error	ROM	Reduced Order Model
SCM	Surrogate Collision Model	SM	Surrogate Model
VM	Virtual Machine	HELSCM	Hybrid Euler-Lagrange Surrogate Collision Model

Bibliography

[1] The National Centre for Research and Development, Fast Track, POIR.01.01.01-00-1253/19, Development and demonstration of a computer system for controlling operation and managing the availability and reliability of industrial infrastructure based on artificial intelligence algorithms (2020-2023) (project budget: 21 079 703.42 PLN), Consortium: Tauron Wytwarzanie S.A. (project promoter), Silesia University of Technology, Sumitomo SHI FW Polska.

[2] ANSYS CFX Inc., [access: 01.02.2021] http://www.ansys.com.

[3] ANSYS® ANSYS, [access: 01.01.2021] https://www.ansys.com.

[4] Bronkhorst High−Tech B.V., [access: 11.02.2021] https://www.bronkhorst.com/int.

[5] CD-adapco, [access: 01.02.2016] http://www.cd-adapco.com.

[6] Cospheric, [access: 11.02.2021] https://www.cospheric.com.

[7] OpenCFD Ltd OpenFOAM, [access: 20.02.2021] http://www.opencfd.co.uk.

[8] Vision Research, Inc., [access: 11.02.2021] https://www.phantomhighspeed.com/products/cameras/veo/veo710.

[9] Adamczyk W.P., Klimanek A., Białecki R.A., Węcel G., Kozołub P., Czakiert T.: Comparison of the standard euler–euler and hybrid euler–lagrange approaches for modeling particle transport in a pilot-scale circulating fluidized bed. „Particuology", 15, 2014, p. 129–137.

[10] Adamczyk W.P., Kozołub P., Klimanek A., Białecki R.A., Andrzejczyk M., Klajny M.: Numerical simulations of the industrial circulating fluidized bed boiler under air- and oxy-fuel combustion. „Applied Thermal Engineering", 87, 2015, p. 127–136.

[11] Adamczyk W.: Application of the Numerical Techniques for Modelling Fluidization Process Within Industrial Scale Boilers. „Arch Comput Methods Eng", 24(4), 2017, p. 669–702.

[12] Adamczyk W., Kozolub P., Wecel G., Klimanek A., Bialecki R., Czakiert T.: Modeling oxy-fuel combustion in a 3D circulating fluidized bed using the hybrid

Euler-Lagrange approach. „Applied Thermal Engineering", 71(1), 2014, p. 266–275.

[13] Adamczyk W., Myöhänen K., Hartge E.U., Ritvanen J., Klimanek A., Hyppänen T., Białecki R.: Generation of data sets for semi-empirical models of circulated fluidized bed boilers using hybrid euler-lagrange technique. „Energy", 143, 2018, p. 219–240.

[14] Adamczyk W., Wecel G., Klajny M., Kozolub P., Klimanek A., Bialecki R., Czakiert T.: Modeling of particle transport and combustion phenomena in a large scale circulating fluidized bed boiler using hybrid Euler-Lagrange approach. „Particuology", 16, 2014, p. 29–40.

[15] Adamczyk W.P.: Application of the Numerical Techniques for Modelling Fluidization Process Within Industrial Scale Boilers. „Archives of Computational Methods in Engineering", 24, 2017, p. 669–702.

[16] Adesina P., O'Sullivan C., Morimoto T., Otsubo M.: Determining a representative element volume for dem simulations of samples with non-circular particles. „Particuology", 68, 2022, p. 29–43.

[17] Aggarwal C.: Neural Networks and Deep Learning: A Textbook 01 2018.

[18] Agler M.T., Wrenn B.A., Zinder S.H., Angenent L.T.: Waste to bioproduct conversion with undefined mixed cultures: the carboxylate platform. „Trends in biotechnology", 29(2), 2011, p. 70–78.

[19] Akiba T., Sano S., Yanase T., Ohta T., Koyama M.: Optuna: A next-generation hyperparameter optimization framework, [in:] Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining 2019, p. 2623–2631.

[20] AL-Ma'amari M. Deep neural networks for regression problems, September 2018.

[21] Alobaid F., Almohammed N., Massoudi Farid M., May J., Rößger P., Richter A., Epple B.: Progress in cfd simulations of fluidized beds for chemical and energy process engineering. „Progress in Energy and Combustion Science", p. 100930, 2021, p. 100930.

[22] Anderson T., Jeckson R.: A fluid mechanical description of fluidized beds. 6(4), November 1967.

[23] Andrews M., O'Rourke P.: The Multifluid Particle-in-Cell (MP-PIC) Method for dense particulate flows. „International Journal of Multiphase Flow", 22(2), 1996, p. 379–402.

[24] Arifuzzaman S., Dong K., Zhu H., Zeng Q.: Dem study and machine learning model of particle percolation under vibration. „Advanced Powder Technology", 33(5), 2022, p. 103551.

[25] Ariyaratne W.K.H., Ratnayake C., Melaaen M.C.: Cfd modeling of dilute phase pneumatic conveying in a horizontal pipe using euler–euler approach. „Particulate Science and Technology", 37(8), 2019, p. 1015–1023.

[26] Arsov N., Pavlovski M., Basnarkov L., Kocarev L.: Generating highly accurate prediction hypotheses through collaborative ensemble learning. „Scientific Reports", 7, 2017.

[27] Aversano G., Ferrarotti M., Parente A.: Digital twin of a combustion furnace operating in flameless conditions: reduced-order model development from cfd simulations. „Proceedings of the Combustion Institute", 38(4), 2021, p. 5373–5381.

[28] Barrasso D., Tamrakar A., Ramachandran R.: A reduced order pbm–ann model of a multi-scale pbm–dem description of a wet granulation process. „Chemical Engineering Science", 119, 2014, p. 319–329.

[29] Belaid M., Falcon R.M.S., Vainikka P.: Pulverized coal versus circulating fluidized-bed boilers - perspectives and challenges for south africa. „South African Journal of Chemical Engineering", 19, 2014, p. 72–81.

[30] Bellman N., Adomian G.: Partial Differential Equations: New Methods for Their Treatment and Solution. Mathematics and Its Applications. Springer Netherlands 1984.

[31] Benyahia S., Galvin-Carney J.: Estimation of numerical errors related to some basic assumptions in discrete particle methods. „Industrial Engineering Chemistry Research", 49, 05 2010.

[32] Bhaskar T., Bhavya B., Singh R., Naik D.V., Kumar A., Goyal H.B.: Chapter 3 - thermochemical conversion of biomass to biofuels, [in:] Pandey A., Larroche C., Ricke S.C., Dussap C.G., Gnansounou E., (eds.), Biofuels, Academic Press, Amsterdam 2011, p. 51–77.

[33] Blaszczuk A., Jagodzik S.: Investigation of heat transfer in a large-scale external heat exchanger with horizontal smooth tube bundle. „Energies", 14(17), 2021.

[34] Braun M., Srinivas M.: Granular flow modeling using R14. „Ansys", 2012.

[35] Breiman L., Friedman J., Stone C., Olshen R.: Classification and Regression Trees. Taylor & Francis 1984.

[36] Breiman L.: Bagging predictors. „Machine Learning", 24(2), 1996, p. 123–140.

[37] Brites I.S.G., da Silva L.M., Barbosa J.L.V., Rigo S.J., Correia S.D., Leithardt V.R.Q.: Machine learning and iot applied to cardiovascular diseases identification through heart sounds: A literature review. „Informatics", 8(4), 2021.

[38] Chapman S., Cowling T.: The mathematical theory of non-uniform gases. Cambridge Univ. Press, Cambridge, 3rd ed., 1970.

[39] Chen H., Rustagi S., Diep E., Langrish T.A., Glasser B.J.: Scale-up of fluidized bed drying: Impact of process and design parameters. „Powder Technology", 339, 2018, p. 8–16.

[40] Chen T., Guestrin C.: Xgboost: A scalable tree boosting system. „CoRR", abs/1603.02754, 2016.

[41] Chen W., Tsutsumi A., Lin H., Otawara K.: Modeling nonlinear dynamics of circulating fluidized beds using neural networks. „China Particuology", 3(1), 2005, p. 84–89.

[42] Chen X., Wang J.: A comparison of two-fluid model, dense discrete particle model and CFD-DEM method for modeling impinging gas–solid flows. „Powder Technology", 254, 2014, p. 94–102.

[43] Cheng S., Langrish T.A.: Fluidized bed drying of chickpeas: Developing a new drying schedule to reduce protein denaturation and remove trypsin inhibitors. „Journal of Food Engineering", 351, 2023, p. 111515.

[44] Claesen M., De Moor B. Hyperparameter search in machine learning, 2015.

[45] Cloete S., Johansen S., Braun M., Popoff B., Amini S.: Evaluation of a Lagrangian discrete phase modeling approach for resolving cluster formation in CFB riser. „7th International Conference on Multiphase Flow", May 2010. , ICMF 2010 Tampa, Floryda USA.

[46] Cloete S., Johansen S.T., Amini S.: Performance evaluation of a complete lagrangian ktgf approach for dilute granular flow modelling. „Powder Technology", 226, 2012, p. 43–52.

[47] Crowe C., Schwarzkopf J., Sommerfeld M., Tsuji Y.: Multiphase flows with droplets and particles. CRC Press, 2 ed., 2012.

[48] Cundall P., Strack O.: A discrete element model for granular assemblies. Géotechnique 1979.

[49] Cyklis P., Młynarczyk P.: The influence of the spatial discretization methods on the nozzle impulse flow simulation results. „Procedia Engineering", 157, 2016, p. 396–403. Selected Papers from IX International Conference on Computational Heat and Mass Transfer (ICCHMT2016).

[50] Dada E.G., Bassi J.S., Chiroma H., Abdulhamid S.M., Adetunmbi A.O., Ajibuwa O.E.: Machine learning for email spam filtering: review, approaches and open research problems. „Heliyon", 5(6), 2019, p. e01802.

[51] Damle A.: 1 - an introduction to the utilization of membrane technology in the production of clean and renewable power, [in:] Gugliuzza A., Basile A., (eds.), Membranes for Clean and Renewable Power Applications, Woodhead Publishing 2014, p. 3–43.

[52] Deutches Institut für Normung. DIN 1942 - Acceptance testing of steam generators, 1994.

[53] Duchi J., Hazan E., Singer Y.: Adaptive subgradient methods for online learning and stochastic optimization. „Journal of Machine Learning Research", 12, 07 2011, p. 2121–2159.

[54] Ergun S.: Fluid Flow through Packed Columns. „Chemical Engineering Progress", 48(2), 1952, p. 89–94.

[55] Farid M., Jeong H., Kim K., Lee J., Kim D., Hwang J.: Numerical investigation of particle transport hydrodynamics and coal combustion in an industrial-scale circulating fluidized bed combustor: Effects of coal feeder positions and coal feeding rates. „Fuel", 192, 2017, p. 187–202.

[56] Fielding R.T.: Architectural styles and the design of network-based software architectures. Publication, University of California, Irvine, 2000.

[57] Gera D., Gautam M., Tsuji Y., Kawaguchi T., Tanaka T.: Computer simulation of bubbles in large-particle fluidized beds. „Powder Technology", 98, 1998, p. 38–47.

[58] Gidaspow D.: Multiphase flow and fluidization. Academic Press, Boston, MA 1994.

[59] Gidaspow D., Huang J.: Kinetic theory based model for blood flow and its viscosity. „Ann. Biomed. Eng.", 37(8), 2009, p. 1534–1545.

[60] Gidaspow D., Jiradilok V.: Computational techniques, the multiphase CFD approach to fluidization and green energy technologies. energy science, engineering and technology series. Nova Science Publishers, Inc., 2nd ed., 2010.

[61] Goldhirsch I.: Introduction to granular temperature. „Powder Technology", 182(2), 2008, p. 130–136. Granular Temperature.

[62] Grochowalski J., Jachymek P., Andrzejczyk M., Klajny M., Widuch A., Morkisz P., Hernik B., Zdeb J., Adamczyk W.: Towards application of machine learning algorithms for prediction temperature distribution within cfb boiler based on specified operating conditions. „Energy", 237, 2021, p. 121538.

[63] Grochowalski J., Widuch A., Sładek S., Melka B., Nowak M., Klimanek A., Andrzejczyk M., Klajny M., Czarnowska L., Hernik B., Zhou M., Pawlak S., Adamczyk W.: Technique for reducing erosion in large-scale circulating fluidized bed units. „Powder Technology", 426, 2023, p. 118651.

[64] Hansen P., Dam-Johansen K., Østergaard K.: High-temperature reaction between sulphur dioxide and limestone—v. the effect of periodically changing oxidizing and reducing conditions. „Chemical Engineering Science", 48(7), 1993, p. 1325–1341.

[65] Hartge E.U., Luecke K., Werther J.: The role of mixing in the performance of cfb reactors. „Chemical Engineering Science", 54(22), 1999, p. 5393–5407.

[66] Heng J., New T., Wilson P.: On the application of an eulerian granular model towards dilute phase pneumatic conveying. „Powder Technology", 327, 2018, p. 456–466.

[67] Hoomans B., Kuipers J., Briels W., Swaaij W.: Discrete particle simulation of bubble and slug formation in a two-dimensional gas-fluidized bed: A hard-sphere approach. „Chemical Engineering Science", 51, 1996, p. 99–118.

[68] Huilin L., Gidaspow D.: Hydrodynamics of binary fluidization in a riser: CFD simulation using two granular temperatures. „Chemical Engineering Science", 58, 2003, p. 3777–3792.

[69] Jain V., Kalo L., Kumar D., Pant H.J., Upadhyay R.K.: Experimental and numerical investigation of liquid–solid binary fluidized beds: Radioactive particle tracking technique and dense discrete phase model simulations. „Particuology", 33, 2017, p. 112–122.

[70] Jenkin J., Savage S.: A theory for the rapid flow of identical, smooth, nearly elastic, spherical particles. „Journal of Fluid Mechanics", 130, May 1983, p. 187–202.

[71] Kaptein M., van den Heuvel E.: Statistics for Data Scientists: An Introduction to Probability, Statistics, and Data Analysis. Undergraduate Topics in Computer Science. Springer International Publishing 2022.

[72] Kelleher J.D., Namee B.M., D'Arcy A.: Fundamentals of Machine Learning for Predictive Data Analytics: Algorithms, Worked Examples, and Case Studies. The MIT Press 2015.

[73] Ketkar N., Santana E.: Deep Learning with Python. O'Reilly 2016.

[74] Kim J., Kang J.: Ai based temperature reduction effect model of fog cooling for human thermal comfort: Climate adaptation technology. „Sustainable Cities and Society", 95, 2023, p. 104574.

[75] Kingma D.P., Ba J. Adam: A method for stochastic optimization, 2014.

[76] Klimanek A., Adamczyk W., Katelbach-Woźniak A., Węcel G., Szlęk A.: Towards a hybrid eulerian–lagrangian cfd modeling of coal gasification in a circulating fluidized bed reactor. „Fuel", 152, 2015, p. 131–137. International Freiberg Conference on IGCC XtL Technologies in 2014.

[77] Klimanek A., Bigda J.: Cfd modelling of co2 enhanced gasification of coal in a pressurized circulating fluidized bed reactor. „Energy", 160, 2018, p. 710–719.

[78] Kloss C., Goniva C.: LIGGGHTS – Open Source Discrete Element Simulations of Granular Materials Based on Lammps, p. 781–788. John Wiley Sons, Ltd 2011.

[79] Korasiak P. Sprawnosc konwersji promieniowania slonecznego na energie elektryczna wspolczesnych ogniw i modulow fotowoltaicznych, March 2017.

[80] Kosowska-Golachowska M., Luckos A.: 28 - direct combustion of microalgae biomass to generate bioelectricity, [in:] Jacob-Lopes E., Zepka L.Q., Severo I.A., Maroneze M.M., (eds.), 3rd Generation Biofuels, , Woodhead Publishing Series in Energy. Woodhead Publishing 2022, p. 665–698.

[81] Ku H.: Notes on the use of propagation of error formulas. „JOURNAL OF RESEARCH of the National Bureau of Standards - C. Engineering and Instrumentation", 70C(4), 1966, p. 263–273.

[82] Kunii D., Levenspiel O.: Fluidization engineering. Butterworth-Heinemann 1991.

[83] Lauriola I., Lavelli A., Aiolli F.: An introduction to deep learning in natural language processing: Models, techniques, and tools. „Neurocomputing", 470, 2022, p. 443–456.

[84] Li Y., Bao J., Yu A., Yang R.: A combined data-driven and discrete modelling approach to predict particle flow in rotating drums. „Chemical Engineering Science", 231, 2021, p. 116251.

[85] Lun C., Savage S., Jeffrey D., Chepurnly N.: Kinetic theories for granular flow: inelastic particles in couette flow and slightly inelastic particles in a general flow field. „Journal of Fluid Mechanics", 140, 1984, p. 223–256.

[86] Ma X., Peng Y., Yu Q., Lai K., Li J., Wu D.: Collision of single particle in rotating flow field. „Chemical Engineering Research and Design", 156, 2020, p. 507–518.

[87] Mcglinchey D., Cowell A., Knight E.A., Pugh J.R., Mason A., Foster B.: Bend pressure drop predictions using the euler-euler model in dense phase pneumatic conveying. „Particulate Science and Technology", 25(6), 2007, p. 495–506.

[88] Messa G.V., Ferrarese G., Malavasi S.: A mixed euler–euler/euler–lagrange approach to erosion prediction. „Wear", 342-343, 2015, p. 138–153.

[89] Miller J.D.: Statistics for Data Science. Packt Publishing 2017.

[90] Mohd Yahya N.S., Ng L.Y., Andiappan V.: Optimisation and planning of biomass supply chain for new and existing power plants based on carbon reduction targets. „Energy", 237, 2021, p. 121488.

[91] Monjaraz-Tec C., Kohlmann L., Schwarz S., Hartung A., Gross J., Krack M.: Prediction and validation of the strongly modulated forced response of two beams undergoing frictional impacts. „Mechanical Systems and Signal Processing", 180, 2022, p. 109410.

[92] Moodley T.L., Govender I.: Experimental validation of dem in rotating drums using positron emission particle tracking. „Mechanics Research Communications", 121, 2022, p. 103861.

[93] Muhammad Ali P., Faraj R.: Data normalization and standardization: A technical report. 01 2014.

[94] Myohanen K., Hyppanen T.: A three-dimensional model frame for modelling combustion and gasification in circulating fluidized bed furnaces. „International Journal Of Chemical Reactor Engineering", 9(A25), 2011.

[95] Myöhänen K., Hyppänen T.: A three-dimensional model frame for modelling combustion and gasification in circulating fluidized bed furnaces. „International Journal of Chemical Reactor Engineering", 9, 2011, p. A25.

[96] Nadda M., Shah S.K., Roy S., Yadav A.: Cfd-based deep neural networks (dnn) model for predicting the hydrodynamics of fluidized beds. „Digital Chemical Engineering", 8, 2023, p. 100113.

[97] Nakao N., Shimamoto H., Yamamoto K.: Cfb combustion control system for multiple fuels. p. 9–15, 03 2011, p. 9–15.

[98] Nikku M., Zhan M., Myöhänen K., Ritvanen J., Li X.: Three-dimensional modeling of a chinese circulating fluidized bed incinerator firing municipal solid waste. „The Journal of Solid Waste Technology and Management", 47(2), 2021, p. 393–405.

[99] Nikku M., Myöhänen K., Ritvanen J., Hyppänen T.: Three-dimensional modeling of fuel flow with a holistic circulating fluidized bed furnace model. „Chemical Engineering Science", 117, 2014, p. 352–363.

[100] Nikku M., Myöhänen K., Ritvanen J., Hyppänen T., Lyytikäinen M.: Three-dimensional modeling of biomass fuel flow in a circulating fluidized bed furnace with an experimentally derived momentum exchange model. „Chemical Engineering Research and Design", 115, 2016, p. 77–90.

[101] Nuortimo K.: State of the art cfb technology for flexible large scale utility power production 2015.

[102] Ogawa S., Umemura A., Oshima N.: On the equation of fully fluidized granular materials. „Journal of Applied Mathematics and Physics", 31, 1980, p. 483.

[103] Ogawa S., Umemura A., Oshima N.: On the Equation of Fully Fluidized Granular Materials. „Journal of Applied Mathematics and Physics", 31, 1980, p. 483–493.

[104] O'Rourke P., Snider D.: An improved collision damping time for MP-PIC calculations of dense particle flows with applications to polydisperse sedimenting beds and colliding particle jets. „Chemical Engineering Science", 65, 2010, p. 6014–6028.

[105] Ouyang Z., Zhu J., Lu Q., Yao Y., Liu J.: The effect of limestone on so2 and nox emissions of pulverized coal combustion preheated by circulating fluidized bed. „Fuel", 120, 2014, p. 116–121.

[106] Pal R.: Chapter 4 - validation methodologies, [in:] Pal R., (ed.), Predictive Modeling of Drug Sensitivity, Academic Press 2017, p. 83–107.

[107] Paska J., Salek M., Surma T. Wywarzanie energii elektrycznej z wykorzystaniem odnawialnych zasobow energii, March 2005.

[108] Patankar N., Joseph D.: Lagrangian numerical simulation of particulate flows. „International Journal of Multiphase Flow", 27(10), 2001, p. 1685–1706.

[109] Patankar N., Joseph D.: Modeling and numerical simulation of particulate flows by the Eulerian-Lagrangian approach. „International Journal of Multiphase Flow", 27(10), 2001, p. 1659–1684.

[110] Poyarkov A., Drutsa A., Khalyavin A., Gusev G., Serdyukov P.: Boosted decision tree regression adjustment for variance reduction in online controlled experiments 08 2016, p. 235–244.

[111] Pronobis M.: Chapter 7 - adaptation of boilers for biomass burning, [in:] Pronobis M., (ed.), Environmentally Oriented Modernization of Power Boilers, Elsevier 2020, p. 177–211.

[112] Rahiala S., Myöhänen K., Hyppänen T.: Modeling the behavior of limestone particles in oxy-fuel CFB processes. „Fuel", 127, 2014, p. 141–150.

[113] Ruder S. An overview of gradient descent optimization algorithms, 2016.

[114] Sahu A.K., Raghavan V., Prasad B.: Temperature effects on hydrodynamics of dense gas-solid flows: Application to bubbling fluidized bed reactors. „International Journal of Thermal Sciences", 124, 2018, p. 387–398.

[115] Sami M., Annamalai K., Wooldridge M.: Co-firing of coal and biomass fuel blends. „Progress in Energy and Combustion Science", 27(2), 2001, p. 171–214.

[116] Schaeffer D.: Instability in the evolution equations describing incompressible granular flow. „Journal of Differentail Equations", 66, 1987, p. 19–55.

[117] Shannon C.E.: A mathematical theory of communication. „The Bell System Technical Journal", 27(3), 1948, p. 379–423.

[118] Sirsat M.S., Fermé E., Câmara J.: Machine learning for brain stroke: A review. „Journal of Stroke and Cerebrovascular Diseases", 29(10), 2020, p. 105162.

[119] Smits A.: A Physical Introduction to Fluid Mechanics. Wiley 2000.

[120] Snider D.: An incompressible three-dimensional Multiphase Particle-in-Cell Model for dense particle flows. „Journal of Computational Physics", 170, 2001, p. 523–549.

[121] Snider D., Banerjee S.: Heterogeneous gas chemistry in the CPFD Eulerian–Lagrangian Numerical scheme (ozone decomposition). „Powder Technology", 199, 2010, p. 100—106.

[122] Snider D., O'Rourke P., Andrews M.: Sediment flow in inclined vessels calculated using a Multiphase Particle-in-Cell Model for dense particle flows. „International Journal of Multiphase Flow", 24, 1998, p. 1359–1382.

[123] Sreekanth P., Syamlal M., O'Brien T.: Computational gas-solids flows and reacting systems: theory, methods and practice. Engineering science reference ed., 2012.

[124] Syamlal M., Rogers W., O'Brien T.: MFIX Documentation. „National Technical Information Service", 1, 1993, p. 5540–5551. Theory Guide.

[125] Szlęk A., Wilk R., Werle S., Schaffel N.: Czyste technologie pozyskiwania energii z węgla oraz perspektywy bezpłomieniowego spalania. 01 2009.

[126] Tavanandi H., Das A., Venkatesh Murthy K., Raghavarao K.: Design and development of a machine for continuous popping and puffing of grains. „Journal of Food Science and Technology -Mysore-", 07 2020.

[127] Topal H., Altınsoy Y., Erbas O.: Design and application of automatic control system in a laboratory scale fluidized bed coal combustion system. 05 2018.

[128] Tsuji T., Yabumoto K., Tanaka T.: Spontaneous structures in three-dimensional bubbling gas-fluidized bed by parallel DEM-CFD coupling simulation. „Powder Technology", 184, 2008, p. 132–140.

[129] Tsuji Y., Kawaguchi T., Tanaka T.: Discrete particle simulation of two-dimensional fluidized bed. „Powder Technology", 77, 1996, p. 79–87.

[130] Vudata Venkata P.B.R.: Prediction of falling solids film thickness near the wall in circulating fluidized bed risers. „International Journal of Chemical Engineering and Applications, Vol. 2, No. 2, April 2011", 04 2011.

[131] Wachem B.V., Schouten J., den Blee C.V., Krishn R., Sinclair J.: Comparative analysis of CFD models of dense gas–solid systems. „AIChE Journal", 47(5), May 2001, p. 1035–1051.

[132] Wang J., Teng H., Han Z., Jensen M.R.: Discrete Element Method in LS-DYNA (DEM). „Livermore", 11 2012.

[133] Wang X.V., Pinter J.S., Liu Z., Wang L.: A machine learning-based image processing approach for robotic assembly system. „Procedia CIRP", 104, 2021, p. 906–911. 54th CIRP CMS 2021 - Towards Digitalized Manufacturing 4.0.

[134] Wang X., Jiang F., Xu X., Fan B., Lei J., Xiao Y.: Experiment and CFD simulation of gas-solid flow in the riser of dense fluidized bed at high gas velocity. „Powder Technology", 199, 2010, p. 203–212.

[135] Wang Z., Liu C., Wang L., Yin S., Tong L.: Dust distribution of solid and adhesive mixed dust in a granular bed filter. „Particuology", 67, 2022, p. 1–7.

[136] Wei H., Zhao Y., Zhang J., Saxén H., Yu Y.: Liggghts and edem application on charging system of ironmaking blast furnace. „Advanced Powder Technology", 28(10), 2017, p. 2482–2487.

[137] Wen C., Yu Y.: Mechanics of Fluidization. „Chemical Engineering Progress Symposium Series", 62, 1966, p. 100–111.

[138] Wheeldon J., Thimsen D.: 13 - economic evaluation of circulating fluidized bed combustion (cfbc) power generation plants, [in:] Scala F., (ed.), Fluidized Bed Technologies for Near-Zero Emission Combustion and Gasification, , Woodhead Publishing Series in Energy. Woodhead Publishing 2013, p. 620–638.

[139] Widuch A., Myöhänen K., Nikku M., Nowak M., Klimanek A., Adamczyk W.: Data set generation at novel test-rig for validation of numerical models for modeling granular flows. „International Journal of Multiphase Flow", 142, 2021, p. 103696.

[140] Widuch A., Myöhänen K., Nikku M., Nowak M., Klimanek A., Adamczyk W.: Towards application of uncertainty quantification procedure combined with experimental procedure for assessment of the accuracy of the dem approach. „International Journal of Multiphase Flow", 142, 2021, p. 103696.

[141] Yassin N.I., Omran S., El Houby E.M., Allam H.: Machine learning techniques for breast cancer computer aided diagnosis using different image modalities: A systematic review. „Computer Methods and Programs in Biomedicine", 156, 2018, p. 25–45.

[142] Ye X., Ni Y.Q., Sajjadi M., Wang Y.W., Lin C.S.: Physics-guided, data-refined modeling of granular material-filled particle dampers by deep transfer learning. „Mechanical Systems and Signal Processing", 180, 2022, p. 109437.

[143] Zhou H., Flamant G., Gauthier D., Lu J.: Lagrangian approach for simulating the gas-particle flow structure in a circulating fluidized bed riser. „International Journal of Multiphase Flow", 28, 2002, p. 1801–1821.

[144] Zhou H., Wang G., Jia C., Li C.: A novel, coupled cfd-dem model for the flow characteristics of particles inside a pipe. „Water", 11(11), 2019.

[145] Zhou L., Song Y., Ji W., Wei H.: Machine learning for combustion. „Energy and AI", 7, 2022, p. 100128.

Internet pages

[146] ANSYS® ANSYS, [access: 01.01.2021] https://www.ansys.com.

[147] Łagisza już w Sieci, [access: 8.08.2023] https://www.cire.pl/pliki/2/lagiszajuzwsieci.pdf.

[148] Jensen A., Fraser K., Laird G.: Improving the precision of discrete element simulations through calibration models 06 2014.

Abstract

Development novel approaches for modeling dense granular flows

Keywords: multiphase flows, particle collisions, machine learning, neural networks, circulating fluidized bed boiler, intelligent control, computational fluid dynamics, experimental setup, simulation validation, novel approach, fluidization

Granular flows, are characterized by a high solid volume fraction of the solid phase. An example of such a flow is the fluidization. Fluidization is a phenomenon where, under the influence of a flowing gas, the solid particles undergo constant motion, and behave like a fluid. The phenomena finds diverse applications, and gained popularity in recent years. It finds it application in industries such as chemicals, pharmaceuticals, food, energy. Especially in the latter example, the technology has numerous applications due to its many advantages over traditional boiler designs. All this, is mainly due to the growing computational resources that allow the use of advanced computer models, to predict the behavior of the solid phase. Due to the large number of particles characterizing granular flows, modeling them is not a trivial task. The large volume solid volume fraction affects a very large number of complex interactions between particles. To reduce the financial outlay accompanying attempts to improve existing technologies, it is necessary to use accurate numerical models to map the phenomena occurring. Despite the computational techniques that are already available, there is still room to try to improve them, or draw advantages from them, to combine them into a new approach.

The dissertation presents two different approaches. One is the use of machine learning techniques to improve collision detection. It is based on combining the Hybrid Euler-Lagrange model with the Discrete Element Method model, through the use of a simplified model. The simplified model will be integrated into simulations using the Hybrid model, through User Defined Functions. When a collision is detected, it will be triggered, thereby replacing the existing approach based on Kinetic Theory of Granular Flow. The results of the calculations will be compared with experimental measurements, carried out on a testing-rig replicating the conditions of the simulations carried out. The second proposed approach uses neural networks to predict field values in a numerical model of a circulating fluidized bed boiler. The approach is part of a prediction and prescriptive system developed, to allow intelligent control of the boiler, located at the Lagisza power plant in Będzin.

Abstrakt

Nowatorskie podejścia do modelowania przepływów granularnych

Słowa kluczowe: przepływy wielowfazowe, kolizje cząstek, uczenie maszynowe, sieci neuronowe, kocioł z cyrkulacyjną warstwą fluidalną, inteligentne sterowanie, komputerowa mechanika płynów, stanowisko eksperymentalne, walidacja symulacji, nowatorskie podejście, fluidizacja

Przepływy granularne, cechują się dużym udziałem objętościowym fazy stałej. Jednym, przykładów takiego przepływu jest zjawisko fluidyzacji. Fluidyzacja jest zjawiskiem, gdzie pod wpływem przepływającego gazu, cząstki stałe są w ciągłym ruchu, co sprawia, iż zachowują się jak płyn. Zastosowanie tego zjawiska jest dość szerokie, a w ostatnich latach zyskuje na popularności. Zjawisto te wykorzystywane jest w przemysłach m.in. chemicznym, farmaceutycznym, spożywczym, energetycznym. Szczególnie w ostatnim z wymienionych przykładów, technologia ta ma liczne zastosowania ze względu na wiele zalet w porównaniu z tradycyjnymi konstrukcjami kotłów. Wszystko to głównie dzięki rosnącym zasobom obliczeniowym, które pozwalają na zastosowanie zaawansowanych modeli komputerowych do przewidywania zachowania fazy stałej.

W związku z dużą ilością cząstek charakteryzującą przepływy granularne, modelowanie ich nie jest trywialnym zadaniem. Duży udział objętościowy wpływa na bardzo dużą ilość złożonych oddziaływań między cząstkami. Aby zmniejszyć nakład finansowy towarzyszący podjętym próbom ulepszenia istniejących technologii, koniecznym jest wykorzystanie dokładnych modeli numerycznych pozwalających na odwzorowanie zjawisk występujących. Mimo dostępnych już technik obliczeniowych, nadal jest miejsce na próby ulepszenia ich lub wyciągnięcia z nich zalet, w celu połączenia ich w nowe podejście.

W pracy doktorskiej zaprezentowane zostały dwa różne podejścia. Jednym z nich jest wykorzystanie technik uczenia maszynowego, w celu usprawnienia wykrywania kolizji. Bazuje ono na połączeniu modelu Hybrydowego Eulera-Lagrange'a z modelem Discrete Element Method, poprzez zastosowanie modelu uproszczonego. Model uproszczony zostanie zintegrowany w symulacjach z użuyciem modelu Hybrydowego, poprzez Funckje Własne Użytkownika, aby w przypadku wykrycia kolizji został on wywołany, czym zastąpi dotychczasowe podejście oparte na kinetycznej Teorii Przepływów Granularnych. Wyniki obliczeń porównane zostaną z pomiarami eksperymentalnymi, przeprowadzonymi na stanowisku, które odwzorowuje warunki przeprowadzanych symulacji. Drugie zaproponowane podejście wykorzystuje sieci neuronowe, w celu predykcji wartości polowych w modelu numerycznym kotła z cyrkulacyjną warstwą fluidalną. Podejście te jest częścią stworzonego systemu predykcji oraz preskrypcji, pozwalającego na inteligentne sterowanie kotłem z znajdującym się w elektrowni Łagisza w Będzinie.