Master's thesis

NTNU
Norwegian University of Science and Technology
Faculty of Natural Sciences
Department of Materials Science and Engineering

Eskil Christensen

Numerical Modeling of Hydrogen Fluoride Adsorption

Master's thesis in TMT 4920 - Materials technology
Supervisor: Kristian Etienne Einarsrud
June 2021

NTNU
Norwegian University of
Science and Technology

Eskil Christensen

Numerical Modeling of Hydrogen Fluoride Adsorption

Master's thesis in TMT 4920 - Materials technology
Supervisor: Kristian Etienne Einarsrud
June 2021

Norwegian University of Science and Technology
Faculty of Natural Sciences
Department of Materials Science and Engineering

NTNU
Norwegian University of
Science and Technology

Preface

This thesis is submitted to the Norwegian University of Science and Technology (NTNU) for completing the degree of Master of Science and Technology.

The work has been conducted at the Department of Materials Science and Engineering at NTNU during the spring semester, between January and July 2021 under the supervision of Professor Kristian Etienne Einarsrud. This thesis includes the theoretical framework and model used to simulate the adsorption of hydrogen fluoride utilizing the computational fluid dynamics software OpenFOAM.

This work would not have been possible to conduct without the supervision and guidance of Professor Kristian Etienne Einarsrud. I am very grateful for all support and guidance you have given me throughout my struggles with mass balances and weird errors I have produced. Your help has been invaluable during the last year.

I would also like to thank my fellow students, especially those who have been seated with me, *We did it!* The last year has been a special one, with a lot of ups and downs, struggles, joy and quarantine. I would like to thank all of you. And to those outside of the university that have believed and supported me. I hope, when you read this, you know this is thank goes to you.

A special thanks go to my family, Arild, Regine and my brother Petter, for always believing in me, for always supporting me, for always inviting me back home when I needed a break, for always being there for me. You might not have understood what I have been ranting on about, but you have always been interested, or pretended to be.
A special thanks goes to the one I long to see and hold, Sandra. This has been a long year, without our talks, the walks home would have been longer and the night darker. Soon the next adventure begins, and I am glad I am starting it with you by my side.

Abstract

The demand for aluminium is increasing annually, aluminium plants increase their cell amperage to increase productivity and economic output, as well as increasing the off-gas temperature. Some plants are installing heat exchangers to reduce the off-gas temperature before it enters the dry scrubber, as an increase in temperature has shown to increase emissions. The model presented in this thesis aims to simulate the effect temperature has on the adsorption of hydrogen fluoride during the initial adsorption when the reaction rate is the rate-limiting step. The model is built in OpenFOAM using the multiphaseEulerFoam solver, considering multiple phases and species. The adsorption rate is modeled with the Langmuir kinetic equation.

The simulation is unstable, most likely due to a poorly defined boundary condition when multiple species are introduced, and an increase in the solid volume fraction is detected. The model generates twice as much mass of adsorbed particles as it should according to the mass balance but can predict trends in the adsorption process with temperature changes. The model needs further work, with the phase transfer term and the assumptions made for the adsorption capacity and reaction rate must be further studied and validated before a finished model using the same approach as the model presented in this thesis. The implemented Langmuir adsorption kinetic equation, that is implemented in the reaction rate coefficient works as intended.

All modified codes and some models are uploaded to GitHub:
https://github.com/EskilC/MasterThesis

Sammendrag

Produksjonsmengden av aluminium øker for hvert år. For å møte etterspørselen, øker aluminiumsverk spenningen i elektrolysecellene for å øke produksjonen, men også den økonomiske gevinsten. En bi-effekt av dette er økt temperatur i avgassene fra elektrolyse prosessen, som øker utslippene av hydrogenfluoride. For å senke temperaturen i avgassene, implementerer industrien varmevekslere. Modellen presentert i denne oppgaven forsøker å simulere effekten temperatur har på adsorpsjonene, i første del av adsorpsjonen hvor den kjemiske reaksjonen er hastighetsbestemmende. Modellen er laget i OpenFOAM for multiphaseEulerFoam som kan håndtere flere faser og stoffer. Absorpsjonshastigheten er beskrevet med en omskrevet Langmuir adsorpsjons kinetikk-lignin.

Simuleringer med denne modellen viser seg å være ustabil, ustabiliteten skyldes mest sannsynlig dårlig definere grensebetingelser når modellene håndterer flere stoffer i fasene, dette vises som en økning i volumet til den solide fasen. Modellen genererer dobbelt så mye masse adsorbert hydrogenfluorid enn det massebalansen tilsier er mulig, men modellen kan fortsatt brukes til å finne trender i adsorpsjonsreaksjonene ved temperaturendringer. Det trengs videre arbeid for å ferdigstille en modell som oppfyller massebalansen. Antagelsene knyttet til adsorpsjons kapasitet og reaksjonshastighet testes og valideres. Implementeringen av Langmuir adsorpsjons kinetikk-ligningen virker som tiltenkt.

Alle modifiserte koder, og enkelte modeller er lastet opp til GitHub:
https://github.com/EskilC/MasterThesis

Table of Contents

List of Figures

List of Tables

1 Introduction

1.1 Aluminium

Aluminium is an essential material in modern engineering, due to its low density, high thermal conductivity, good corrosion resistance, and is easily cast, machined and formed. The aluminium production has had an annual growth of 2.5% on average over the last 25 years, and in 2019 the production was 64 million metric ton [1]. While the industry is growing, some predict the production might peak before 2050 [2]. Primary aluminium production involves two energy-intensive processes to transform the raw material, bauxite, to metallic aluminium, the Bayer and Hall-Héroult process.

The Bayer process transforms bauxite ore into smelting grade alumina (SGA). Process parameters in the Bayer process affect parameters like: impurity levels, particle size, specific surface area, pore size, moisture and hydroxyl content [3]. The alumina is transformed into molten aluminium with the Hall-Héroult process in electrolytic cells, sketch of the electrolytic cell is shown in figure 1.1. The electrochemical reaction in the Hall-Héroult process is written as:

$$\frac{1}{2}\mathrm{Al_2O_{3(s)}} + \frac{3}{4}\mathrm{C_{(s)}} = \mathrm{Al_{(l)}} + \frac{3}{4}\mathrm{CO_{2(g)}} \tag{rx.1}$$

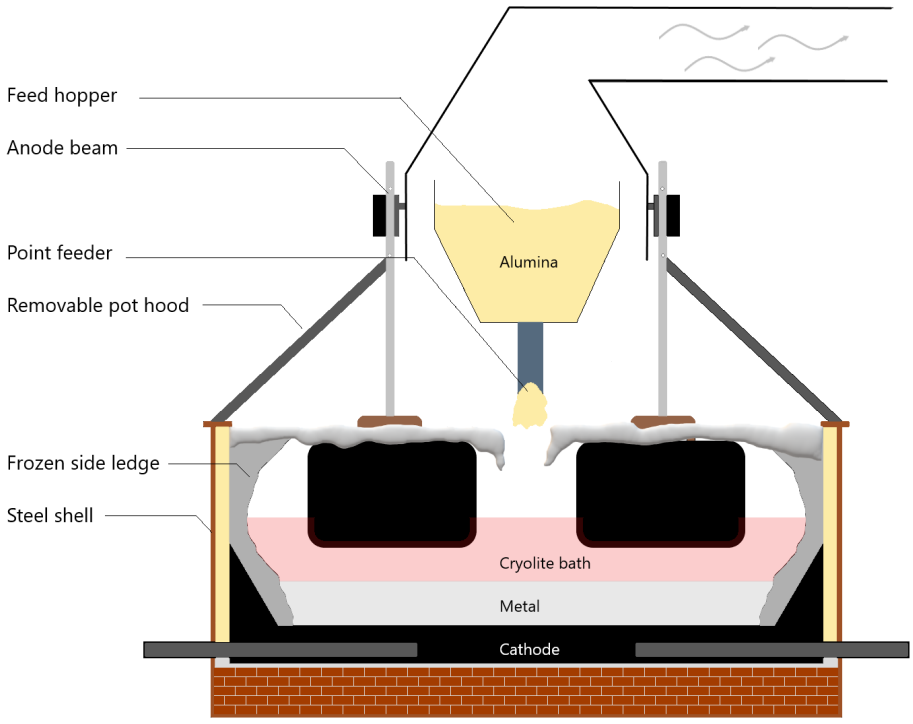


Figure 1.1: Schematic of an aluminium reduction cell [4].

Alumina is dissolved in an electrolyte consisting of cryolite (Na_3AlF_6), aluminum fluoride (AlF_3) and fluorite (CaF_2). The alumina content in the electrolyte is usually 2-3 wt% alumina. A lower concentration of alumina might lead to the so-called anode effect, causing a big increase in the cell voltage, whereas a higher concentration of alumina may lead to an excessive amount of undissolved alumina [5].

The undissolved alumina laying on top of the self-formed, frozen crust, formed on the surface of the electrolyte, will act as an isolating lid, as well as a physical barrier keeping ambient moisture away from the electrolyte. The alumina is feed into the electrolyte in portions through feeder holes made in the crust. An electric current, from the anode to the cathode, passes through the electrolyte and reduces the alumina according to reaction rx.1. The molten aluminium settles at the bottom due to the difference in density and the molten aluminium is collected from the cell regularly.

The aluminium process relies heavily on fluoride, as it offers sufficient dissolution of alumina, as well as good electric conductivity. The drawback of using fluoride is the generation and emissions of hydrogen fluoride, and as the industry seeks to increase productivity and economic returns, the cell amperage increase with the inevitable increase in hydrogen fluoride production by [6];

- More frequent unscheduled anode changes.
- Increased temperature in the off-gas.
- Increase rates of crust collapse.

1.2 Hydrogen Fluoride Generation

Hydrogen fluoride (HF) is a greenhouse gas, but the biggest risk associated with emissions of HF is the effect it can have on people, animals and vegetation [7]. HF gas and particulates emitted into the atmosphere can deposit on the surface of vegetation and dissolves into the apoplast, affecting the photosynthetic process, causing injury and ultimately affect the growth and yield of vegetation [8]. In humans, HF is an acute poison that interferes with the calcium metabolism and can cause system toxicity, cardiac arrest, and in worst case, death. Short-term exposure can lead to irritation and severe respiratory damage and/or irritation of the eyes [9].

HF is generated in small amounts in the reduction process of alumina to aluminium. The primary generation of HF in the aluminium process is from electrochemical evolution taking place within the electrolyte. HF is formed by a reaction between hydrogen-containing species and the electrolyte, where the most favored reactions are reaction rx.2 and rx.3 [10–12].

$$2\,Na_3AlF_{6(l)} + 3\,H_2O = Al_2O_{3(s)} + 6\,NaF_{(s)} + 6\,HF_{(g)} \quad \text{(rx.2)}$$

$$2\,AlF_{3(l)} + 3\,H_2O = Al_{2O3(s)} + 6\,HF_{(g)} \quad \text{(rx.3)}$$

The secondary generation of HF is through thermal hydrolysis of the pot gas. The most favored reaction is reaction rx.4. Figure 1.2 gives an overview of the different sources of hydrogen and to what form of generation they contributes too. Loss on ignition (LOI) and moisture on ignition (MOI) are terms used to differentiate between the two forms of water that can be found in alumina. MOI, also referred to as LOI 25-300, is the adsorbed water that is highly volatile. LOI 300-1000 refers to the structural water, usually in the form of residual hydroxyl from gibbsite [10].

$$3\,NaAlF_{4(g)} + 3\,H_2O_{(g)} = 6\,HF_{(g)} + Al_2O_{3(s)} + Na_3AlF_{6(l/s)} \qquad \text{(rx.4)}$$

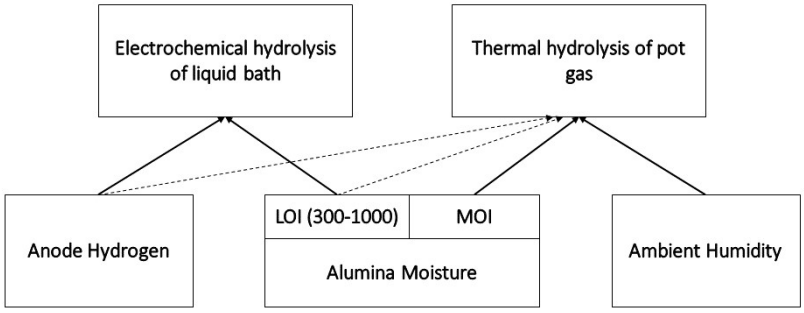


Figure 1.2: Overview of the different sources of hydrogen and their contribution to different HF generation mechanism, recreated from [10].

Table 1.1 shows the amount of HF generated from the different hydrogen sources. The secondary generation, in table 1.1, does also include the generation from particulates at the surface as well as the desorption of fluoride from the alumina feed. The alumina feed consists of primary alumina and secondary alumina, where secondary alumina is alumina that has been used in the dry scrubber to clean the off-gas from the electrolytic cell.

Table 1.1: Overview of the contribution from the different HF generation sources, recreated from [12].

Emissions Contributor	Emission (kg/tonne Al)	HF Emission (%)
Primary generation		
Alumina moisture	7-14	40-60
Anodes hydrogen	2-5	< 10
Dissolved water	3-10	10-30
Secondary generation		
Hydrolysis	2-8	10-30
Fluoride desorption	0.1-0.5	< 1

Most of the generated fluoride particulate is the result of vaporization of the electrolyte [13]. Electrolyte enters the off-gas from bursting of bubbles at the surface of the electrolyte, where small droplets of the liquid electrolyte are entrained, forming particulates containing fluoride as the off-gas cools. The crust on the electrolyte acts as a filter, removing entrained electrolyte, as well as increasing the distance it must travel to escape, increasing the chance of the entrained electrolyte settles [13].

Generally, the concentration of gaseous HF in the off-gas is 110-350 $\mathrm{mg\,Nm^{-3}}$ [3], depending on the alumina quality, alumina feeding, hooding efficiency, crust integrity, and humidity to mention a few parameters. The generation of HF is a problem for the aluminium industry, as emission not only poses an environmental issue but also an economical loss as the lost fluoride needs to be replaced to maintain the fluoride concentration in the bath [14]. The solution is to clean the gas by capture the HF and re-introduce the captured fluoride into the melt. SGA, which is the feedstock for the electrolytic cell, is suitable to capture HF due to its high surface area and a favorable affinity for fluoride [3]. The adsorption of fluoride and capture of particulates takes place in the dry scrubber, which is a part of the gas treatment center (GTC). The dry scrubber outputs a combination of fluoride enriched alumina, also referred to as secondary alumina or reacted alumina, and fluoride-rich particulates, by some called bath fines. Figure 1.3 shows the fluoride circulation path in aluminium production.

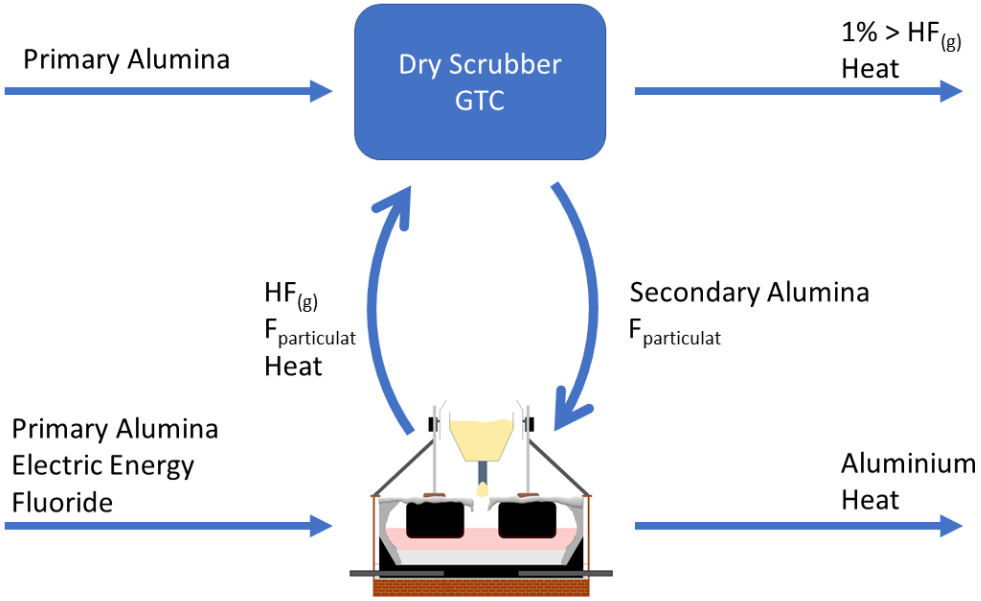


Figure 1.3: Shows the hydrogen fluoride circle in the aluminium process.

1.3 Gas Treatment Center and Scrubber

The scrubbing process is a unit operation in which one or more component of a gas is selectively absorbed by an absorbent. Scrubbing is commonly encountered when cleaning flue gas to control acid gases, particulates, etc. [15].

At an aluminium plant, individual reduction cells are hooded to collect the off-gas generated in the electrolytic reduction process. The individual pot exhaust ducts are combined, transporting the off-gas to the GTC to clean the off-gas from harmful substances [16]. The number of pots that are connected to a single unit in the GTC can be as many as 200 [17], making the gas concentration in the GTC the average concentration of all pots. This results in the GTC receiving a steady stream of off-gas with a small deviation in the HF concentration, and consequently the secondary alumina leaving the GTC having an even fluoride loading. The GTC can consist of two different scrubbers, dry scrubbers and wet scrubbers. The dry scrubbers used at alumina plants are usually based on a two-step process. The first being the adsorption of HF on alumina in the gas stream and the second adsorption in the bag filter where alumina and fluoride particulates are stopped by the bag filters.

The second scrubber found in some GTCs is the wet scrubber. Generally, only some plants have a wet scrubber, whose main task is to reduce the sulfur emissions, primary in the form of sulfur dioxide SO_2 [18]. Wet scrubbers are mainly utilized in Scandinavian countries [19] as they require a large amount of water, seawater is commonly used. Seawater naturally contains sodium bicarbonate ($NaHCO_3$) that is a natural reagent for SO_2. The produced sodium sulfate (Na_2SO_4) is also naturally present in seawater. While some plants might have to transport the water over a long distance, the plants in Norway are usually located close to fjords or the coast, giving them the advantages of having intakes and outlets close to the plant. If the seawater does not need any alteration in temperature or oxygen level it can be discharged directly into the sea [18].

The cleaning efficiency of GTC is commonly $\geq 99\%$, meaning that plants with modern GTCs release about 0.03-0.06 kg HF per ton produced alumina [20]. In recent years the temperature of the off-gas has increased, resulting in the off-gas needing cooling due to material limitations of the bag filters and operation efficiency of the scrubber. One solution utilized is cooling the off-gas with water, but an increased focus on heat recovery has driven research on utilizing heat exchangers. How the heat exchangers affect the dry scrubber has been investigated by looking at the flow velocity and flow profile, as well as measuring the concentration when the off-gas enters the dry scrubber at a decreased temperature. There is still a lot to investigate to fully understand the positive benefits of decreasing the inlet temperature of the off-gas.

1.4 Aim and Goal

The scrubber design has been improved upon, but the principals are the same now, as it was in the 60's. The implementation of heat exchangers downstream from the dry scrubber can allow for optimization of the dry scrubber by controlling the temperature of the off-gas entering the scrubber. There is currently, to the authors knowledge, no available computational fluid dynamics model that can describe the adsorption of an gaseous specie on a solid in the Eulerian-Eulerian frame of reference.

The purpose of this work is to develop a computational fluid dynamics model that can simulate the adsorption of HF taking place in the dry scrubber found at aluminium plants. The hope is to inspire the continuous development into a complete model that industry can utilize to optimize the scrubbing process of off-gas.

This thesis has six goals:

- Give the reader insight into the adsorption process and the effect temperature has on the process.
- Present a set of equation describing the fluid flow and reaction.
- Introduce the reader to computational fluid dynamics.
- Present modification done on source code in the multiphaseEulerFoam.
- Verify the new model with mass balance calculations.
- Highlight difficulties and propose future work.

The reader will through this thesis gain a basic knowledge of the aluminium production, the generation of hydrogen fluoride, dry scrubber function and design, methods used in numerical modeling and simulations. The reader will gain access to the main codes used to generate the model, and the different parameters used in each of the simulations.

2 Literature Review

2.1 Gas Treatment Center

Most of GTC in aluminium industry utilizes dry scrubbers to clean the off-gas. Dry scrubbers utilized in the cleaning of HF containing off-gas released from the electrolytic cell are based on the adsorption of HF on alumina particles.

2.2 Adsorption

Adsorption is an exothermic process that takes place on the surface, or interface, of a bubble or solid. The adsorbed specie is usually referred to as adsorbate and the specie where the adsorption takes place the adsorbent. Theoretically, adsorption can be divided into two different adsorption mechanisms, depending on the nature of the bond between the adsorbent and adsorbate. If the bond is physical, e.g. due to van der Waals forces, the adsorption reaction is called physical adsorption or physisorption. If instead, the bond is due to chemical forces, i.e transfer of electrons, the adsorption is called chemisorption. Adsorption can form a single layer (monolayer adsorption) or multiple layers (multilayer adsorption), as well as being desorbed, shown in figure 2.1.

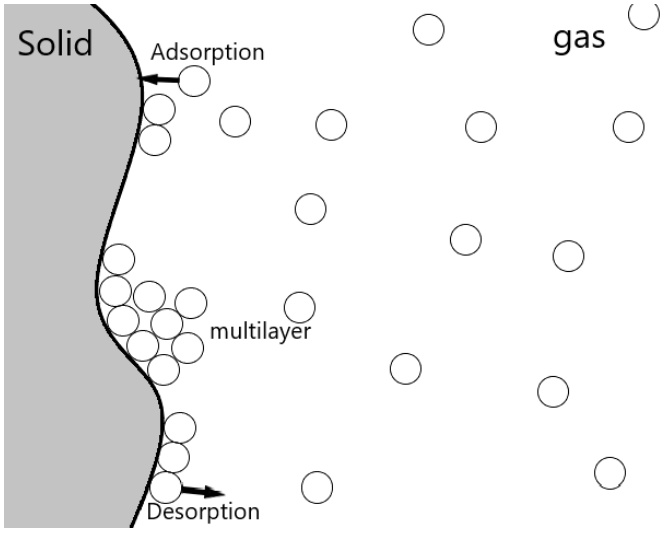


Figure 2.1: Show adsorption, desorption and the formation of mono- and multilayers on a solid surface.

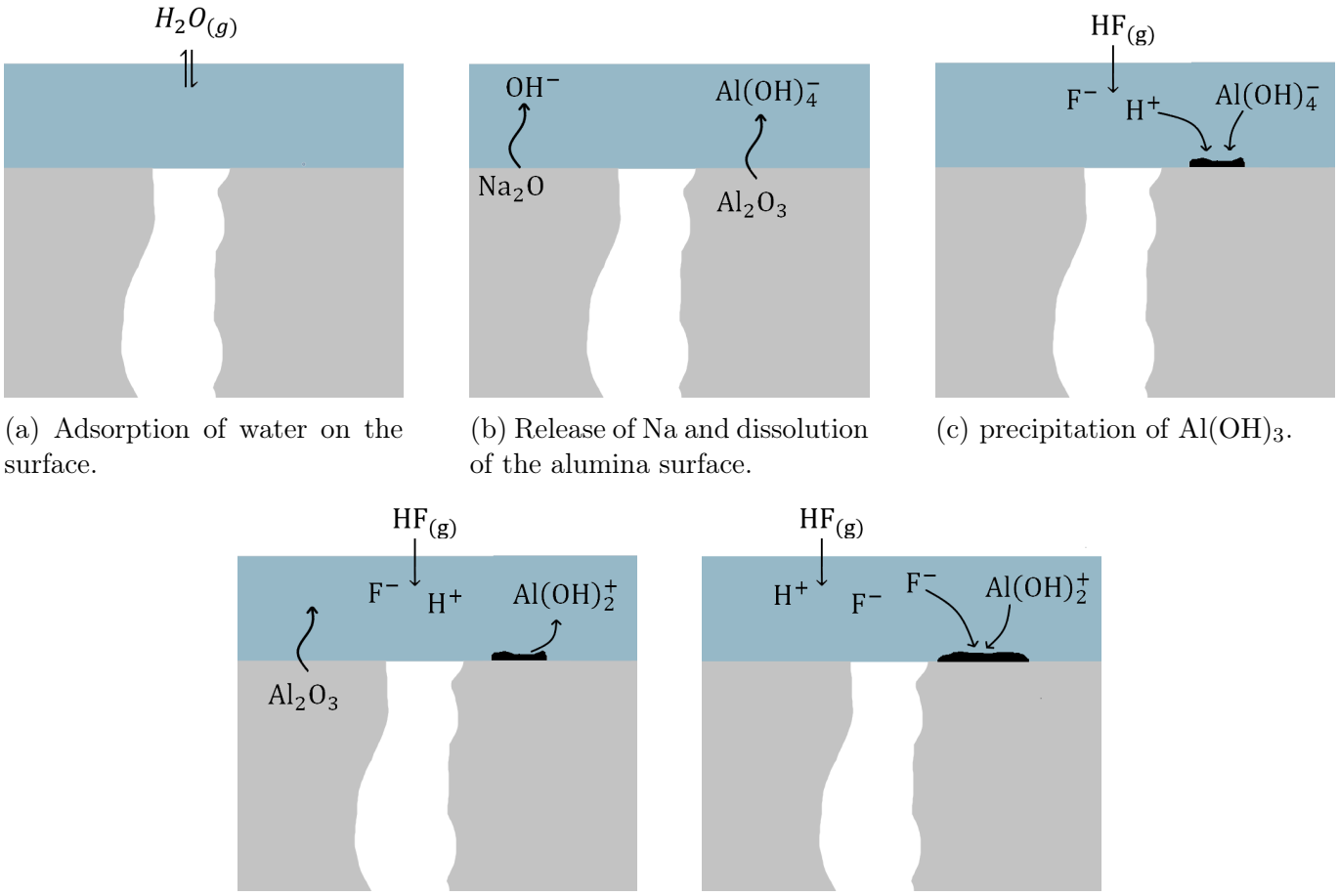
Physical adsorption is essentially exothermic, but the reaction of gas with the surface layer of a solid may lead to the formation of endothermic compounds. Because of this, chemisorption may have an endothermic character [21]. The adsorption of HF on alumina, being the basis for modern dry scrubbers, has been investigated to determine the adsorption mechanism [22–24], where the most widely accepted mechanisms are those proposed by Gillespie [25] and Dando et al. [26].

2.2.1 Adsorption mechanism

Gillespie [25] suggested in 1992 two different adsorption mechanism, one for low relative humidity ($RH_{low} < 35\%$) and one for high relative humidity ($RH_{high} > 35\%$). The adsorption of HF was dependent on the reversible adsorption of water, which was found to be an important step in Gillespie's proposed mechanism. For RH_{low} the fluoride capacity was proportional to the relative humidity, ranging from 0.4 - 1.7 $mg\,m^{-2}$, Gillespie proposed the following adsorption mechanism:

- Water is rapidly physisorbed and forms a layer on the alumina surface. The thickness of the layer is determined by the relative humidity.
- Sodium on the surface may be released into the aqueous layer, increasing the local pH, which causes the alumina to dissolve as $AlO_2{}^-$ and/or $Al(OH)_4{}^-$.
- HF is adsorbed into the aqueous layer, the pH returns to neutrality and small quantities of $Al(OH)_3$ precipitates.
- Continued adsorption of HF into the aqueous layer decreases the pH bellow neutrality, leading to the alumina surface dissolves to form $Al(OH)_2{}^+$ and $Al(OH)^{2+}$. These ions, then react with the fluoride in the aqueous layer and $AlF_x(OH)_{3-x} \cdot {}_6H_2O$ precipitates.
- The reaction stops when the local water is saturated with $AlF_x(OH)_{3-x} \cdot {}_6H_2O$.

As the aqueous layer increases in thickness with relative humidity, the quantity of reaction product needed to saturate it increases, therefore increasing the adsorption capacity, figure 2.2. For RH_{high} the adsorption capacity was independent of relative humidity and was found to be 3.6 $mg\,m^{-2}$. Gillespie proposed that the water condensed inside the pores, compared to RH_{low} where water condensed only on the surface, increasing the amount of water on the particle, thus increasing the adsorption capacity, figure 2.3.



(a) Adsorption of water on the surface.

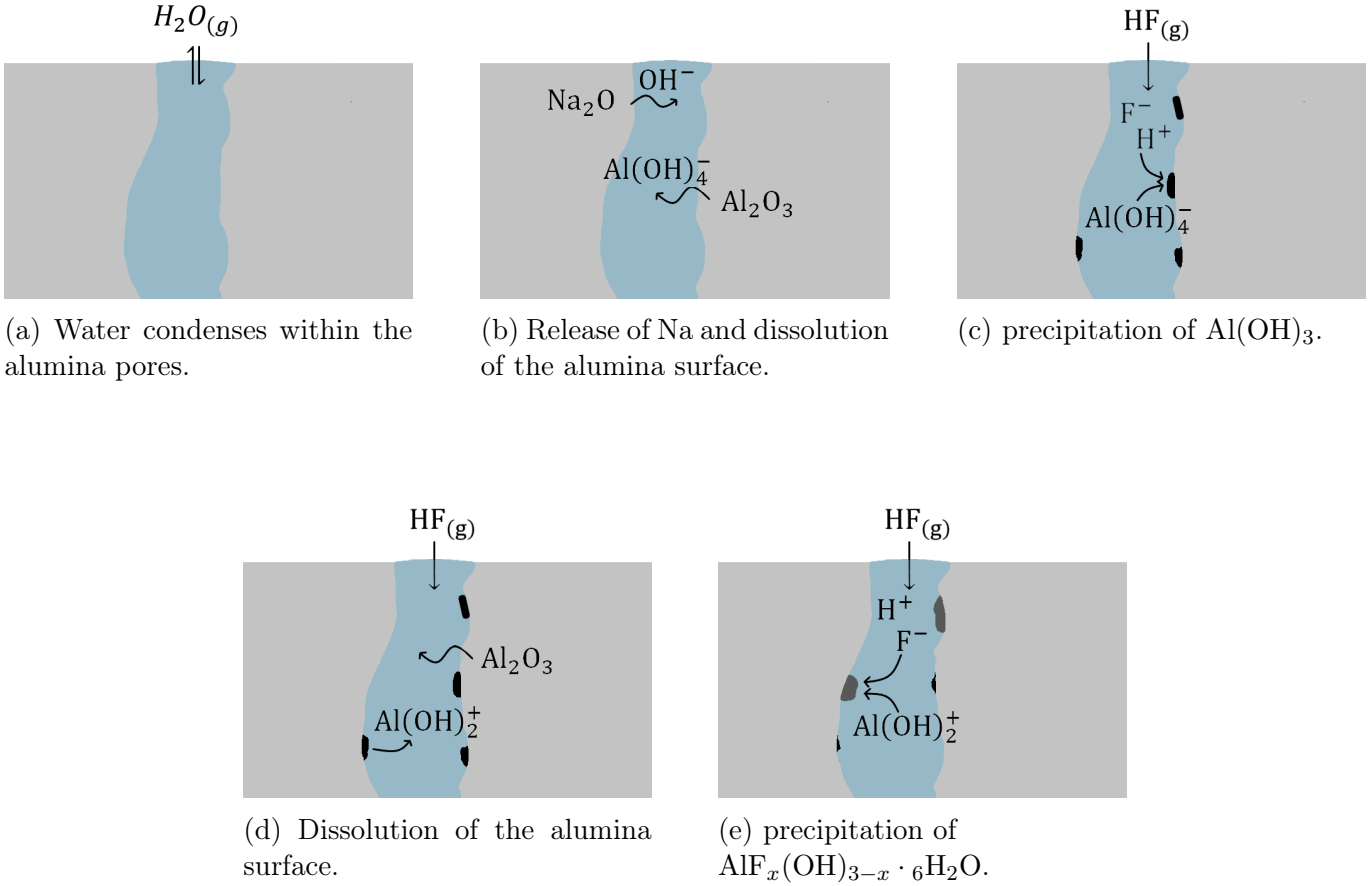
(b) Release of Na and dissolution of the alumina surface.

(c) precipitation of $Al(OH)_3$.

(d) Dissolution of the alumina surface.

(e) precipitation of $AlF_x(OH)_{3-x} \cdot {}_6H_2O$.

Figure 2.2: The figure show the model proposed by Gillespie of $AlF_x(OH)_{3-x} \cdot {}_6H_2O$ formation on smelter grade alumina under low relative humidity conditions, from [4].



(a) Water condenses within the alumina pores.

(b) Release of Na and dissolution of the alumina surface.

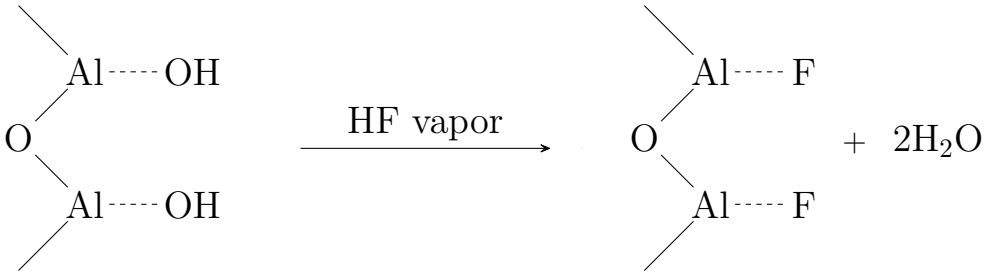
(c) precipitation of $Al(OH)_3$.

(d) Dissolution of the alumina surface.

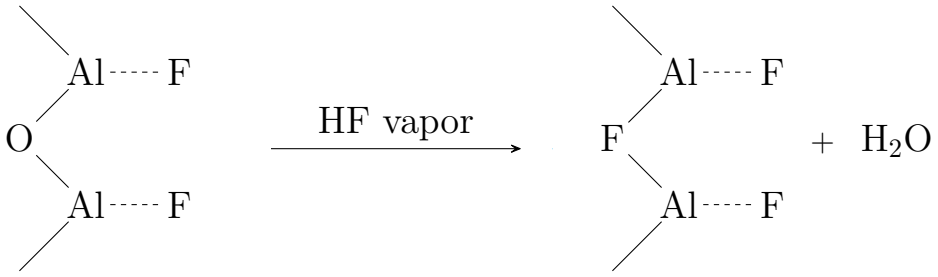
(e) precipitation of $AlF_x(OH)_{3-x} \cdot {}_6H_2O$.

Figure 2.3: The figure show the model proposed by Gillespie of $AlF_x(OH)_{3-x} \cdot {}_6H_2O$ formation in smelter grade alumina under high relative humidity conditions, from [4].

Dando et al. [26] proposed the following adsorption mechanism after finding three different chemical forms of adsorbed fluoride on the surface of the fluorinated alumina. They found AlF_3, $AlF_3 \cdot xH_2O$ (x in range 1-3) and AlOF (terminal oxyfluoried), figure 2.4a, proposing that there were formed non-bridging Al-F bonds on the alumina particle surface by fluoride replacing the hydroxyl group, figure 2.4b. After this finding, they concluded that the adsorption, i.e formation of AlF_3 and $AlF_3 \cdot xH_2O$, depends on the alumina's surface initial degree of hydration. The local growth of the Al-F phase was believed to be dependent on the amount of physisorbed moisture, the moisture on ignition, temperature, and the extent of surface converted to AlF_3. This conversion was limited to $<2\%$ due to the short time in the reactor.



(a) shows the formation of terminal oxyfluoride



(b) Shows the formation of bridging fluoride

Figure 2.4: The figures shows the reaction mechanisms behind the adsorption of HF on smelting grade alumina suggested by Dando et al. [26].

2.2.2 The influence of temperature

The trend towards higher amperage in the electrolytic cells results in a higher pot gas temperature entering the GTC [27, 28]. It is therefore essential to have an understanding of the effect temperature has on the adsorption.

Earlier investigations done on the HF emission and its temperature dependence have shown that HF emissions increase with temperature. Figure 2.5 shows the measured HF emission, done by Qassab et al. [29], after the bag filters when the gas temperature was increased from 110 to 120 ℃. Sørhuus and Ose [27], found a similar dependency when comparing emissions during the hot season in the middle east to emissions in Norway. They postulated that part of the increased emission was due to regeneration of HF from particulates trapped in the filter bags and that the temperature influenced the regeneration more than humidity.

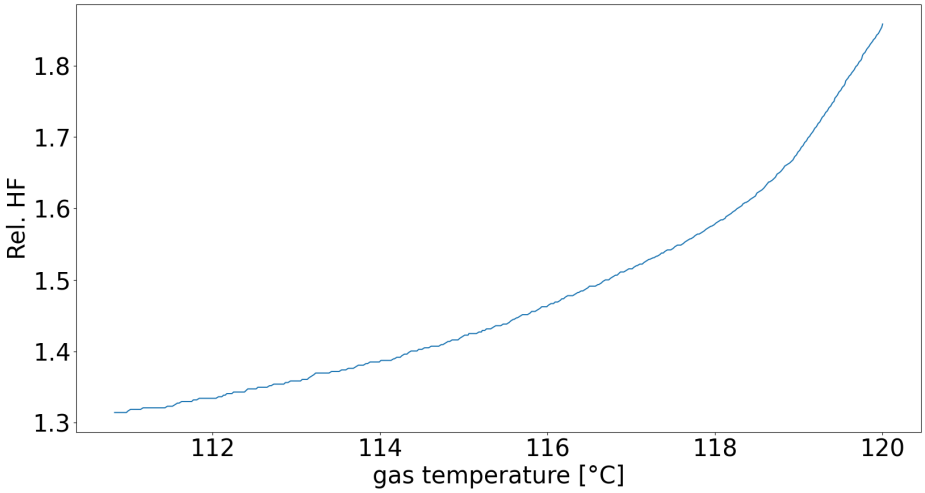


Figure 2.5: Shows the HF emission with varying gas temperatures measured from the filters, re-created from [29].

Bonnier et al. [30] measured the stack emission during a thunderstorm, while the ambient temperature decreased and ambient humidity increase. They reported a drop in the emission during the storm. The emission returned to normal levels after the storm.

Agbenyegah [3], linked the phenomena of increased emissions at higher temperatures with increasing reaction rate. The increased temperature increases the chemical reaction rate, which in turn increases the rate of pore blocking. Pore blocking on alumina during fluorination has been observed earlier [31, 32]. McIntosh et al. [33], made a mathematical model predicting the pore size distribution after fluorination.

Agbenyegah [3] obtained the following basic regression model from the response on surface data on the fluoride loading at breaktrough ($\boldsymbol{F_{bt}}$) in wt%F, temperature in ℃ and humidity (H) in g water/kg gas:

For a HF concentration of 320 mg Nm^{-3}:

$$\boldsymbol{F_{bt}} = 3.31 - 0.0133T - 0.0113H \tag{1}$$

and for HF concentration of 650 mg Nm^{-3}:

$$\boldsymbol{F_{bt}} = 2.98 - 0.0104T - 0.0109H \tag{2}$$

The pore blocking as the product of adsorption, hydroxyfluoride, introduces microstructural constraints inside the alumina pores, subsequently completely blocking the pore, or reduces the gas diffusion into the pore. This again promotes reaction to take place on the external alumina surface. At elevated temperatures, this process occurs at a faster rate. Agbenyegah [3] measured the surface

area, with Brunauer-Emmett-Teller (BET) [34] and Barrett–Joyner–Halenda (BJH) [35] method, of the alumina particles during fluorination, figure 2.6.

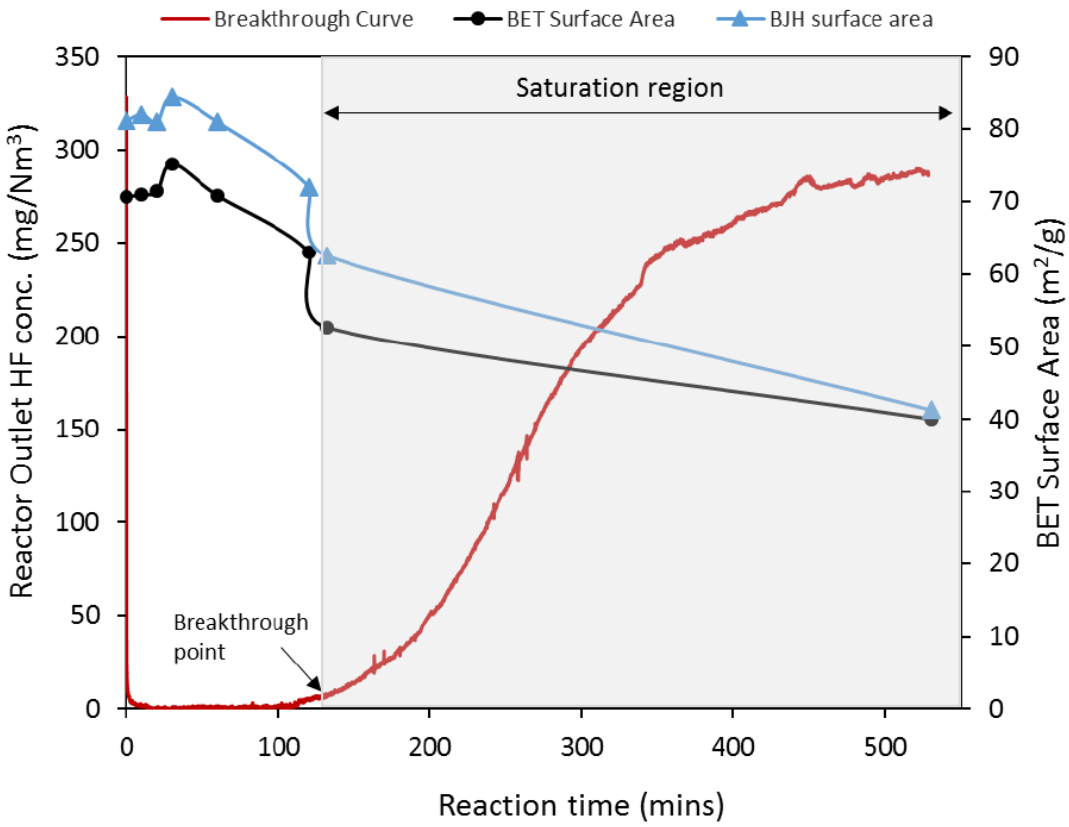


Figure 2.6: Shows the evolution of the surface area during fluorination for both BET and BJH surface area, from [3].

The reduction in surface area was believed to be because of pore blocking, but the increase in surface area was not explained. It is therefore believed that pore blocking is a major factor when it comes to the kinetics of HF adsorption.

2.2.3 Desorption of hydrogen fluoride

The desorption of HF from secondary alumina has been investigated earlier as a part of determining the adsorption isotherm [22, 36, 37]. All three experiments found that hydrogen fluoride is not desorbed at temperatures below 400 °C. This lead to the conclusion of HF adsorption being irreversible at operating temperatures in the dry scrubber.

2.2.4 Adsorption kinetics

Agbenyegah [3] determined the reaction rate ($\dot{R}$) by thermometric titration, assessing the depletion of acid and base sites during HF capture. He estimated the total adsorption reaction rate to $2.5 \cdot 10^{-8}$ $\mathrm{mol\,g^{-1}\,s^{-1}}$ by titration of 0.5 g alumina reacted with 320 $\mathrm{mg\,Nm^{-3}}$ HF at 100 °C.

The rate of reaction decreases over time for irreversible adsorption, as the concentration of reactive sites decreases. Elovich formulated the Elovich equation [38], equation 3, to describe the chemisorption process, where q is the amount of gas adsorbed, α_r is the initial adsorption rate and α_E is the relative rate of active site decay.

$$\frac{dq}{dt} = \alpha e^{-\alpha_E q} \tag{3}$$

The integrated form of equation 3, with the boundary condition of $q(t = 0) = 0$, with $t_0 = \frac{1}{\alpha_r \alpha_E}$ [39]:

$$q = \frac{2.3}{\alpha_E} log(t + t_0) - \frac{2.3}{\alpha_E} log(t_0) \tag{4}$$

The Elovich equation 3, has shown to deviate from experimental observation toward the end of the adsorption [39] when compared to experimental data.

Agbenyegah [3] conducted an experiment to test the applicability of the Elovich model on the adsorption of HF. He calculated the active site decay (α_E) to 0.037 and 0.028 g $mg^{-1}\,min^{-1}$ (t_0=100) for the pre- and post-breakthrough periods, respectively. The difference in active site decay was assumed to be due to a change in the adsorption mechanism. He concluded that the adsorption of HF goes through three stages. First, the adsorption rate is controlled by the rate of the chemisorption reaction. Secondly, the rate is both reaction and diffusion-limited, also know as intermediate controlled, until it reaches the breakthrough point. After the breaktrough, the rate is purely diffusion-limited.

Yang et al. [40] investigated the adsorption of water on the alumina. Even though the water adsorption is physisorption, they found that water shows a similar trend as HF. A higher adsorption temperature indicates a faster adsorption rate but subsequently reduces the equilibrium adsorption capacity.

2.3 Dry Scrubber

One of the important factors for a good adsorption process is high contact area between the adsorbate and adsorbent. Therefore the majority of GTC facilities utilize injection reactors, or a form of fluidized bed reactors to ensure a high contact surface area [3]. One of the dry scrubber technologies in use is the ABART (Aluminium Best Available Recovery Technologies) from General Electric, sketch of the dry scrubber is shown in figure 2.7. ABART utilizes the counter-current flow to maximize the utilization of adsorption capacity of the alumina particles. The first alumina to react with the HF containing off-gas is alumina that has been used further up in the dry scrubber. The relatively high concentration of HF at the inlet is subsequently adsorbed throughout the scrubber as it carries the small alumina particles upstream. The increased cross-sectional area in the bag filter room reduces the flow rate and the alumina falls down and is collected and transported to the electrolytic cell. The off-gas has a lower concentration of HF when it reaches the bag filters, to further adsorb fluoride, fresh alumina is injected into the filters to adsorb as much fluoride as possible. The filters trap

particulates as well as restricting alumina fines from entering the wet-scrubber. The gas is vacuumed into the secondary scrubber located upstream from the bag filters.

The off-gas can have a temperature exceeding 120 ℃ when it reaches the GTC. There has been conducted research on heat exchanges, and how to best extract excess heat with different configurations and locations of the heat exchanger [43, 44]. The issue with transporting alumina and fluoride particulates from the scrubber to the electrolytic cell is the segregation. Sørhuus et al. [17] propose a pot integrated scrubber based on the ABART system patented by General Electric. The concept is that every pot gets a scrubber that is located much closer to the pots than the centralized GTC, thereof the name Pot Integrated Abart (PIA). If the PIA is going to work, the cooling of the off-gas may become very important, to ensure low operation temperature in the dry scrubber. The off-gas entering the dry-scrubber have usually been transported for some time, usually resulting in the off-gas being cooled 10-15 ℃ before entering the GTC [43].

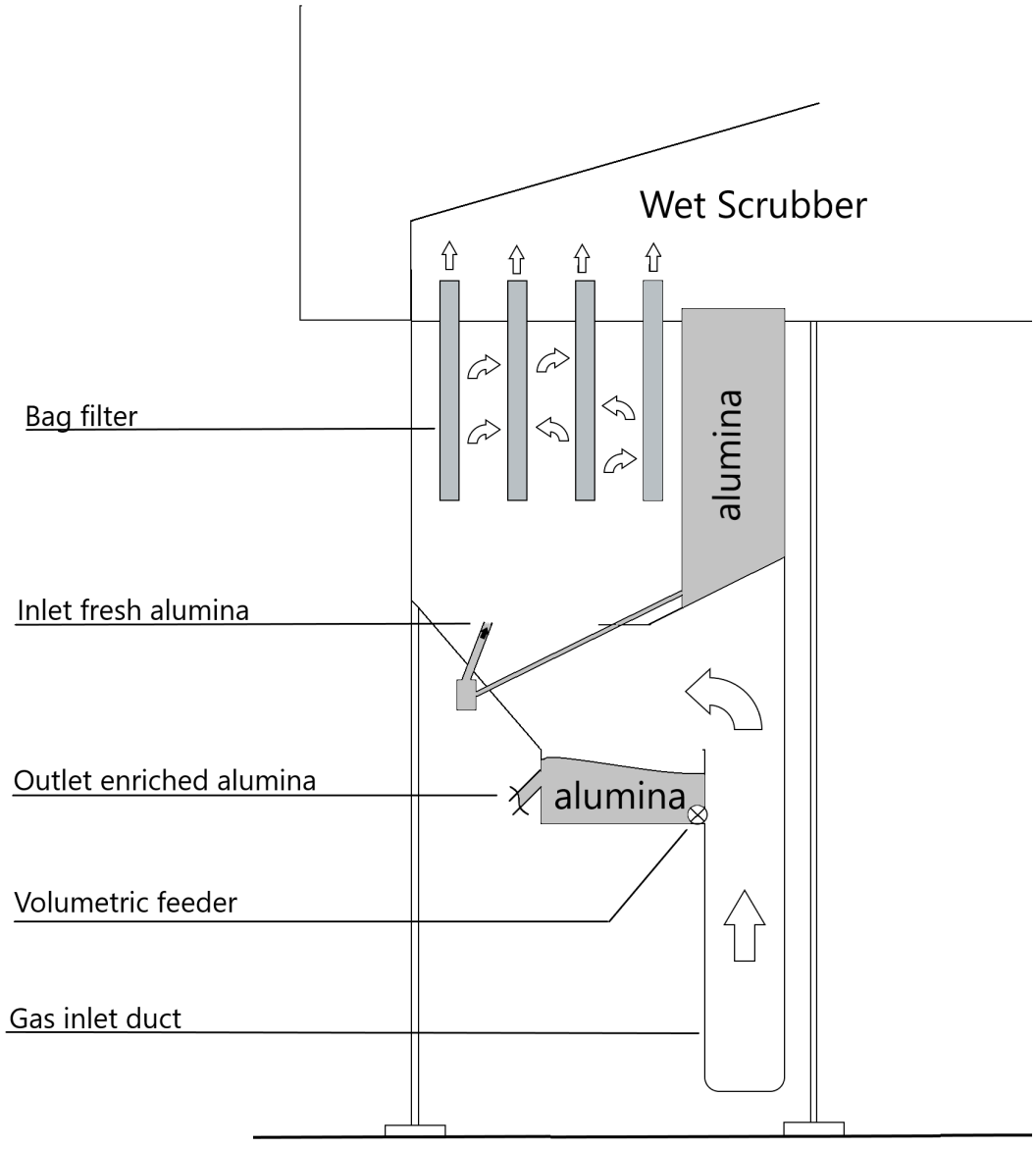


Figure 2.7: Sketch of the dry-scrubber in a gas treatment center, generated from [41] and [42].

2.3.1 Heat exchangers

In addition to the earlier explained reason for reducing the off-gas temperature due to operating temperature and increased emissions, a secondary reason exists. The aluminium industry is already a high energy-intensive process, so much that it accounts for around 3-4 % of the world's total electricity consumption. About half of the energy inputted is lost to the surroundings in the form of heat [45]. The aluminium industry has for a long time investigated and invested in the recovery of waste heat. The off-gas from the electrolytic cells can have a temperature exceeding 120 °C when entering the GTC. The total heat lost in the off-gas can account for up to 45% of the loss related to the electrolytic cell [45]. Sørhuus et al. [43], found that it was possible to recover 25 MW of waste heat from one pot line.

The heat exchanger might affect other parameters than just the temperature. An integrated heat exchanger might affect the velocity profile, making it more uniform, increasing flow distribution between multiple scrubbers due to the resistance introduced by the heat exchanger [42].

The alumina industry faces specific challenges when using heat exchangers to recover waste heat from low-temperature sources like the off-gas. In the off-gas, there are acidic gases, water vapor, particulates that can lead to extensive corrosion, or fouling [46]. The fouling occurs mainly because of the off-gas containing particulates and moisture. The fouling process can occur in the ducts transporting the off-gas, the heat exchangers, as well as the pipes transporting fluoride enriched alumina, forming hard gray scales inside ducts and transport pipes. The mechanism behind the formation of the hard gray scales is qualitatively understood [16]. The formation of hard gray scales will reduce the effect of heat exchangers if they are formed [47], as well as negatively impact the maintenance and efficiency of the dry scrubber [48].

2.4 Bed Reactors

Gas-solid fluidized bed reactors are extensively used in dry scrubbers as well as the process industry because of their advantageous properties. Fluidized beds typically are more complex in design and operation compared to other reactors, such as packed bed and stirred tank reactors, but it is said to be three distinctive advantages compared to the other reactors:

- Superior heat and mass transfer.
- Solids move like a fluid (fluidized).
- Allows for a wide range in particle size.

For most exothermic reactions, a fluidized bed can maintain an isothermal profile within a few degrees [49].

2.4.1 Packed bed

The packed bed reactors belong to the most widely applied reactors, due to their simplicity, effectiveness, low capital and operating costs. The reactants flowing through the packed bed can be a

gas, liquid, or both. In a packed bed the fluid flow rate is so low that the particle bed does not expand as the fluid merely percolates through the space between the solid particles, therefore the bed is referred to as a packed bed or fixed bed.

The shape and size of the particles in a packed bed are determined by the characteristics of the individual processes, to achieve high effectiveness, at a reasonable cost, as to utilize the particles and the reactor volume. In the case of a porous solid particle, e.g. alumina, the largest portion of the surface area consists of pore walls. For a given conversion rate, the external surface determines the flux density of the reactant to the internal surface area of the pore walls [50]. The specific surface area [m^2 kg^{-1}], i.e ratio between external surface area and particle volume, should be as high as possible to achieve an overall higher reactive surface area per reactor volume. This is achieved by decreasing the particle size, or by modifying the particle shape.

By altering the particle size or shape, the packed bed porosity might change as different shapes have different packing factors, leading to a difference in the void fraction (ϵ). A packed bed with smaller particles will have a smaller void fraction compared to larger particles. The void fraction is defined as the ratio of the volume the fluid phase occupies to the volume of the bed, as shown in equation 5.

$$\epsilon = \frac{V_{fluid}}{V_{bed}} = \frac{V_{fluid}}{V_{fluid} + V_{particles}} \tag{5}$$

The pressure drops over the packed bed due to friction between the particles and the fluid can be characterized by the linear dependence upon the flow velocity and the inertia (Forcheimer effect)[50]. The Ergun equation, equation 6, combines these two contribution to calculate the pressure drop (Δp) as a function of bed height (H), fluid flow velocity ($u4$), kinematic viscosity (μ), fluid density (ρ_f) and particle diameter (d_p)[51].

$$\frac{\Delta p}{H} = \frac{150\mu(1-\epsilon)^2}{d_p^2\epsilon^3}u + \frac{1.75\rho_f(1-\epsilon)}{d_p\epsilon^3}u^2 \tag{6}$$

or

$$\frac{\Delta p}{H} = \frac{150\mu(1-\epsilon)^2}{d_p^2\epsilon^3 K_{fs}}u + \frac{1.75\rho_f(1-\epsilon)}{d_p\epsilon^3 K_{fs}}u^2 \tag{7}$$

K_fs is the the inter-phase exchange coefficient between the fluid and particles. The pressure drop increase with increasing flow velocity, towards a critical velocity called minimum fluidization velocity. When the flow velocity reaches the minimum fluidization velocity, the particle bed is said to be fluidized, meaning that the upward force is equal or greater than the downward forces.

2.4.2 Fluidized bed

Fluidization is said to happen when a fluid flowing through a bed of particles with a velocity exceeding the minimum fluidization velocity, but not so great that the particles get carried away in the ascending flow. The particles used in fluidized bed reactors are usually classified into four types, depending on their fluidization behavior. This was first done by Geldart [52], figure 2.8 shows Geldarts diagram for the different particle classification A, B, C and D. The different particles behaves differently during fluidization.

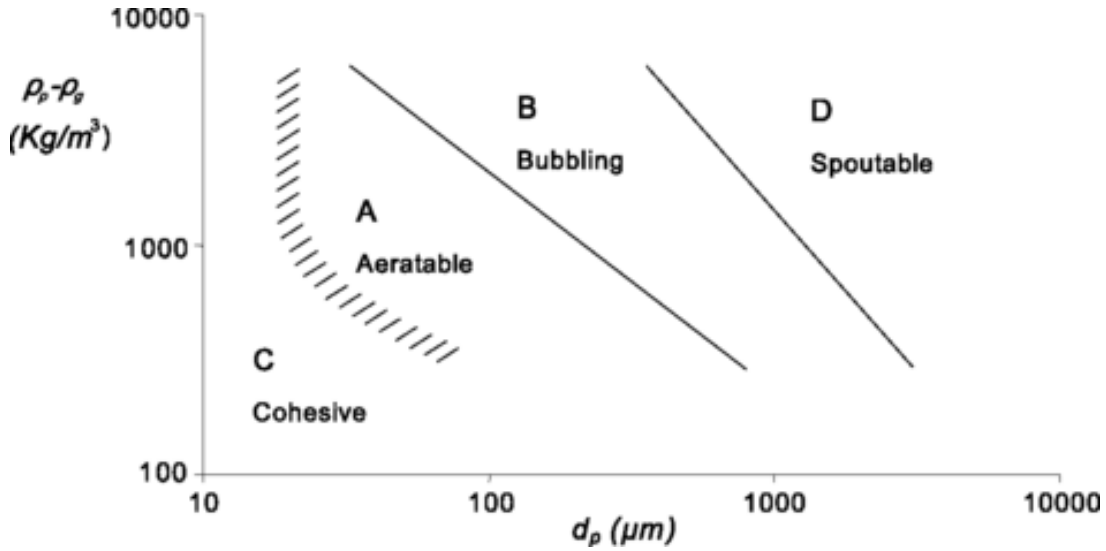


Figure 2.8: Shows Geldart's diagram for the different classification of powder, from [53].

When the bed becomes fully fluidized, the bed reaches minimum fluidization velocity, the pressure drop does not increase with fluid velocity, as the pressure drop is solely due to the weight of the suspended bed [49]. This pressure drop can be estimated with equation 8 [54]:

$$\Delta p = H(1 - \epsilon_f)(\rho_p - \rho_f)g \tag{8}$$

where H is the bed height, ϵ is the void fraction, ρ is density and g is gravitational acceleration. The two-phase theory describes the flow inside the fluidized bed, and can be expressed as;

$$\dot{V}_{bed} = \dot{V}_{emulsion} + \dot{V}_{bubbles} = A_{bed}u_{fm} + A_{bed}(u_0 - u_{fm}) \tag{9}$$

where $\dot{V}_{bed}$ is the total volumetric flow rate, $\dot{V}_{emulsion}$ and $\dot{V}_{bubbles}$ is the total volumetric flow rate through the particle phase and bubbles, A_{bed} is the cross-sectional area of the bed, u_{fm} is the minimum fluidization velocity and u_0 is the superficial gas velocity. The two-phase theory does not account for the smooth fluidization regime or the transport of gas to or from the bubbles. The fluidized bed can be divided into different degrees or steps of fluidization, figure 2.9. When the velocity increases beyond the minimum fluidization velocity, bubbles can form. Small and light particles formes smooth fluidization before bubbles appear, were bigger or denser particles tend to form bubbles at the point of fluidization. Further increase of the fluid velocity leads to the fluidized bed becoming turbulent, and the gas voids in the bed become elongated. Further increase in fluid flow velocity will lead to the particle being carried out of the bed [49].

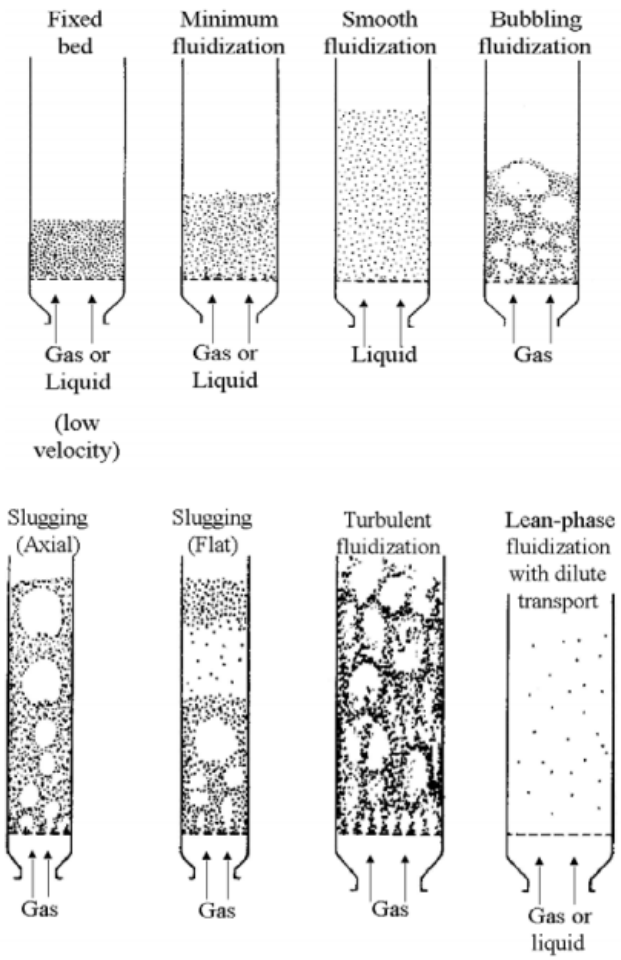


Figure 2.9: The various types of fluidized beds, figure from [55].

The bubbles coalesce as they rise in the bed and if the ratio of the height of the bed to the diameter of the bed, the diameter of the bubbles can become close to that of the bed, this is referred to as slugging [56].

Vasconcelos and Mesquita [57] investigated the minimum fluidization velocity for alumina used in the alumina industry at different size fractions. They found that the pressure drop increased with increasing fluid velocity until the transition from fixed bed to fluidized bed occurred. From the transition point until fully fluidized, the pressure drop continued rising at an increased rate.

2.5 Previous Models and Simulations

2.5.1 Multiphase flow

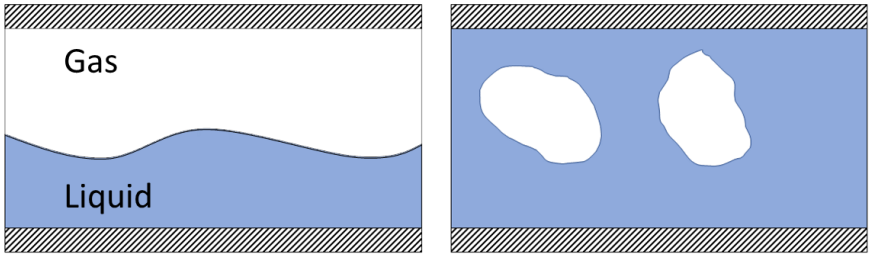
Multiphase flow is the kind of flow that occurs most frequently in nature and technology. A phase should be understood in the thermodynamic sense, as a solid, liquid, or gas-like state that can exist in a one- or multi-component system [58]. Multiphase flow can be divided into two different classifications, separated flows and dispersed flows, figure 2.10. This classification is important from a physical point of view, and also the computational perspective, as the approach for simulating separated flows and dispersed flows are different. Dispersed flow is characterized by the flow, where one phase is dispersed in a continuous phase, i.e. carrier phase. In a gas-solid or liquid-solid flow, the solid is always the dispersed phase. In gas-liquid or liquid-liquid flows, the dispersed flow phase is determined mainly by the flow rate of both phases. The phase with the highest flow rate is in most cases the continuous phase [59].

In order to simplify multiplephases systems, a concept of phase fractions is introduced. Phase fraction are sometimes referred to as volume fraction or void fraction, and the definition vary in literature. One intuitive definition is that the total volume occupied by phase k, V_k divided by the total volume, V_{tot};

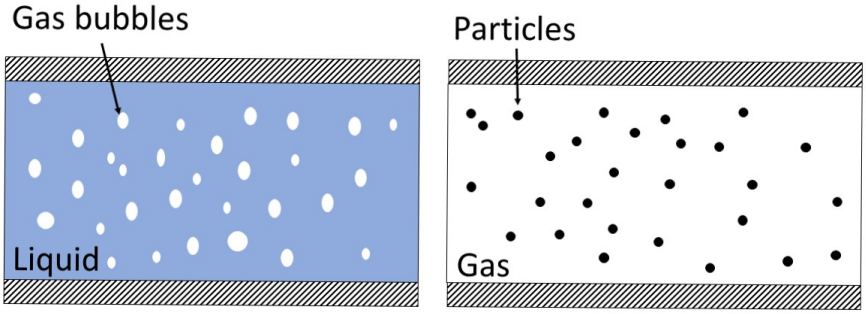
$$\alpha_k = \frac{V_k}{V_{tot}} \tag{10}$$

Crown [60] defines the volume fraction of the phase as the volume of the dispersed phase, δV_k, in volume δV, when δV approaches a volume δV^0, which is the limiting volume that ensures a stationary average;

$$\alpha_k = \lim_{\delta V \to \delta V^0} \frac{\delta V_k}{\delta V} \tag{11}$$



(a) Separated two-phase flow



(b) Dispersed two-phase flow

Figure 2.10: Shows the difference between the different multiphase flows.

Numerical simulations, based on governing principles of mass and momentum transfer, are utilized to fundamentally investigate phase interactions inside different equipment, e.g. reactors, pipes, or storage tanks. This makes it possible to connect material properties and process conditions to measured results, without empiricism towards predictive design and operation.

For dispersed flow, there are different computational strategies, distinguished based on the scales resolved by the model formulation [59]. Multiphase flow is usually modeled using two different approaches; the Lagrangian-Eulerian and Eulerian-Eulerian models. Both of the approaches consider the fluid phase (gas or liquid) as the continuous phase, where the fundamental difference between them is how the solid phase is treated. The Lagrangian-Eulerian models describe the solid phase on a particle level, where Newton's law is applied to describe the movement of the particles. The Eulerian-Eulerian models treat the solid phase as a continuum, more details on multiphase flow in chapter 3.2.

2.5.2 Fluidized bed simulations

Liu and Hinrichsen [61] implemented a two-fluid solver in OpenFOAM in the Eulerian-Eulerian frame of reference. They validated the solver with two bubbling fluidization bed cases, one with uniform gas feed and the other with a central gas jet, figure 2.11, by comparing them to experimental data from literature. Two different drag models were utilized, for the uniform gas feed simulation the Syamlal model was used, and Gidaspow's model for the central gas jet simulation. The 2D computation

domain for the uniform gas feed consisted of 11.200 rectangular cells, corresponding to a grid width of 0.005 m. The uniform gas feed bed had an inlet superficial gas velocity of 0.38 and 0.46 m/s. Their results were calculated from the last 55 seconds of 60 seconds simulations. They reported that their simulated results of solid volume fraction are in good agreement with the experimental data from the literature.

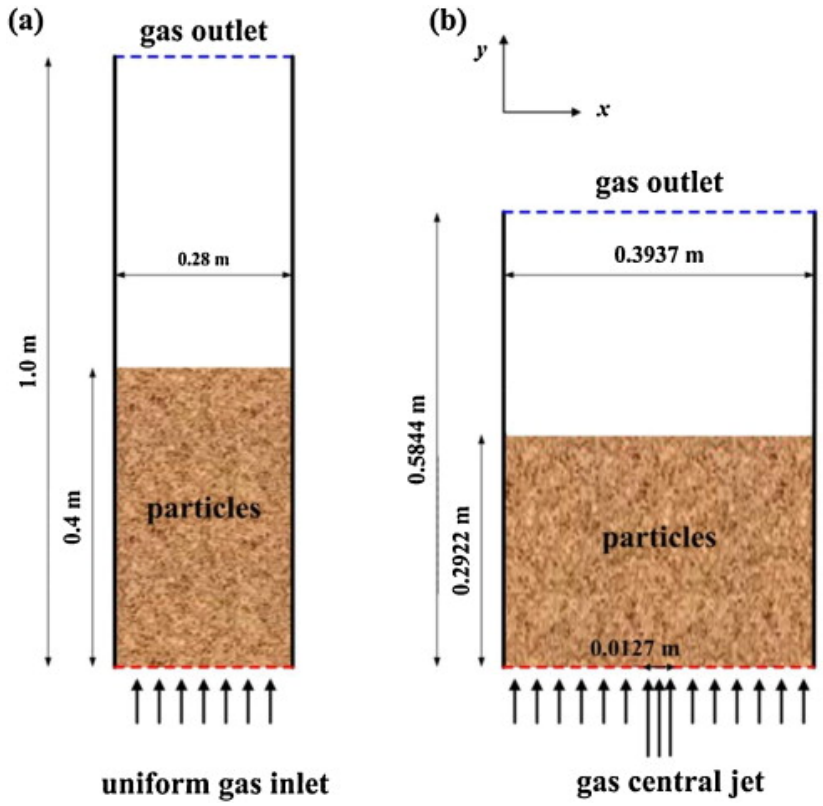


Figure 2.11: Schematics of fluidized beds with (a) uniform gas feed and (b) a central jet, from [61].

Herzog et al. [62] did a comparative study of different computational fluid dynamics (CFD) codes for simulations of gas-solid fluidized bed hydrodynamics. One of the models they compared was an OpenFOAM multi-fluid Eulerian-Eulerian model (twoPhaseEulerPimpleFoam from OpenFOAM v2.0), where the solid-phase properties were calculated with the kinetic theory of a granular flow. Momentum exchange coefficients were calculated with Gidaspow and Symlal-O'Brian drag models. The grid was made out of 11200 rectangular cells, corresponding to a width of 5 mm, and the time step was set to 10^{-5} second. They confirmed the findings by Taghipour et al. [63], that the simulation reaches statistically steady-state behavior after 3 seconds.

Li et al. [64] investigated the drag closures in spout fluidized beds using OpenFOAM. They used a discrete phase model (DPM) and CFD in a DPM-CFD coupled model, i.e Lagrangian-Eulerian framework. The work was based on the DPMEFoam solver with some modifications to the collision model and the implementation of drag closures. The particle used in the simulation was a 6mm teflon bead, and they found that with a grid size smaller than the particle diameter their model predictions were inaccurate. When comparing their simulations to an experiment conducted by Bokkers [65],

they adopted Bokkers simulation of packing of bed. Then, by comparing the height of the packed bed in the experiment to the height in the simulated bed, finding the number of particles in their system.

Venier et al. [66] compared ANSYS Fluent v19.2 and OpenFOAM v6.0 with respect to their Euler-Euler kinetic theory of granular flow, and compared the results from the simulations to experimental results for Geldart A, B and D particles. The only inter-phase force considered in the simulations was the drag force term, which was defined by the K_{sp} coefficient which relates the drag force with the relative velocity of both phases. For Geldart B and D particles, a modified Gidaspow drag model was used (modified by the sphericity of the particles). Geldart A particles used a modified Gibilaro drag model to calculate the drag coefficient. They concluded with both ANSYS Fluent and OpenFOAM providing a fair prediction of the bubble and solid fractions for freely-bubbling Geldart B and D particles, but there is room for improvement with the more demanding Geldart A particles. Both programs gave an accurate description of the fluidization curve, after accounting for the particle sphericity. In an earlier study Venier found that the Johnson-Jackson boundary condition [67], with a high specularity coefficient, roughly predicts the general hydrodynamic behavior of particle-wall interaction with good accuracy.

Passalacqua and Fox [68] used the Neumann boundary condition for the pressure at the inlet, on the outlet the pressure was set to atmospheric pressure, and on the walls, zeroGradient was used. The no-slip B.C was used on the walls for the fluid phase and Johnson -Jackson for the particle phase. The simulation was able to predict the behavior of their system with time steps in the range $10^{-5} - 10^{-4}$ s. Their model was capable of describing a system with steep gradients and it was stable when complete phase separation or a dense, nearly packed, particle phase was present.

2.5.3 Multiphase reaction simulations

Hlawitschka et al. [69] developed and validated a multiphase Eulerian-Eulerian solver to analyze the chemisorption of CO_2 in a rectangular column. Their model includes an interface sharpening term, equation 12, in the continuity equation.

$$\nabla[\alpha_l \boldsymbol{u}_{comp}(1-\alpha_l)] \tag{12}$$

where $\boldsymbol{u}_{comp}$ is the interface compression velocity and α_l is the liquid holdup, i.e volume fraction. In their simulation they included virtual mass force and drag forces, and the turbulence modeling was based on LES (Large Eddy Simulations). The adsorption term, i.e. adsorption rate, appeared in the specie transport equation, equation 13, where $\dot{m}^j$ is the mass transfer term and $\dot{S}^j$ the production term;

$$\frac{\partial}{\partial t}(\alpha_l \rho_l Y^j) + \nabla \cdot (\alpha_l \rho_l \boldsymbol{u}_l Y^j) - \nabla \cdot (\alpha_l \rho_l D^j \nabla Y^j) = \dot{m}^j + \alpha_l \dot{S}^j \tag{13}$$

The mass transfer term was described with equation 14, where d is the bubble diameter, E is the enhancement factor, α_i is the gas phase volume fraction, D^j is the diffusion coefficient, Sh is the Sherwood number and H is the Henry constant.

$$\dot{m}^j = \frac{6}{d^2} E \alpha_i D^j Sh^j \rho_l (H^j \frac{\rho_g}{\rho_l} - Y_l^j) \tag{14}$$

The production term was calculated as an ordinary chemical reversible reaction, where the temperature-dependent reaction rate was calculated from a modified Arrhenius equation. The simulation showed a good hydrodynamic behavior, a small time delay between the simulation results and results from literature was found. This time delay was believed to be due to bubble size and the enhancement factor E.

Dupre et al. [70] investigated the possibility of using semi-coke waste as a sorbent for SO_2 in fixed bed columns. They used CFD-modeling to characterize the effect adsorption capacity and rate have on the reactor design. They used the Eulerian-Eulerian method, and the k-ϵ model to describe the turbulence in the gas phase. The inter-phase momentum transfer came from the drag force, which was calculated using Gidaspow's model. The mass transfer was calculated from the following equation;

$$\frac{\partial}{\partial t}(\alpha_i \rho_l) + \boldsymbol{u}_i \nabla(\alpha_i \rho_l) = \nabla^2 \rho_l + \dot{q} \tag{15}$$

Where i denotes the phase, l is the specie in phase i, and $\dot{q}$ is the time derivative of the specie of the adsorbate inside the particle. $\dot{q}$, was determined experimentally. The units in the equation used to calculate the mass transfer do not add up, as a diffusion coefficient is missing in the first term on the left side.

Panicker et al. [71] simulated the aluminium process using OpenFOAM. They used a mesh with cells that had a size of 15 mm and the particle diameter was 10^{-4} m. They used degassing, no-slip, free slip for the velocity boundary conditions, zeroGradient for the specie, voltage, magnetic field, volume fractions, and turbulent boundary conditions. They compared their model's prediction of alumina dissolution with experiments from literature and found that it matched *reasonably well*.

Yang et al. [72], presented a multiphase OpenFOAM model capable of simulate transport and reactions inside microscale reactors. They state that the major challenge of simulating mass transfer in a multiphase system is the discontinuity created by the concentration difference across the phase interfaces. To solve this, they implemented an additional term, ϕ_j, representing a one-field formulation proposed by Haroun et al. [73], that only takes none zero value in the interface region.

3 Mathematical Basis

3.1 Governing Equation

Conservation laws can be derived by considering a given quantity of matter, i.e a control mass or material volume (m.v), and its extensive properties, such as mass, momentum and energy [74]. In fluid flows, it is more convenient to use a control volume instead of a control mass.

Consider an arbitrary finite region fixed in space inside a flow field, where the closed surface (c.s) defines the control volume (c.v). dS is a surface element made out of a finite area on c.s, and it is associated, outward-pointing, unit normal vector $\vec{n}$, figure 3.1.

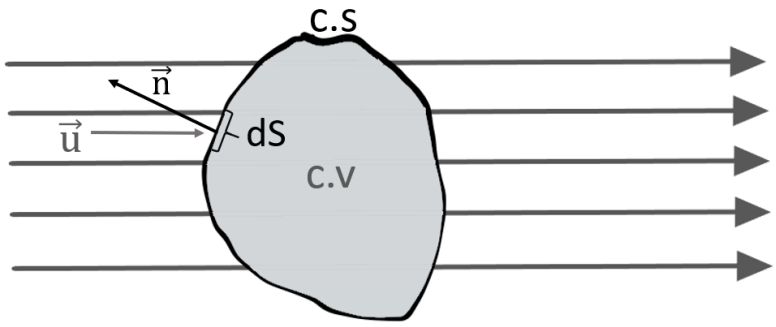


Figure 3.1: Definition of a finite, fixed in space, control volume inside a flow field

3.1.1 Conservation equation

The conservation equations are derived from conservation laws for extensive properties, i.e. mass or momentum. The conservation law relates the rate of change of extensive property to externally determined effects [74]. The conservation equation for mass, which can neither be destroyed or generated, is written as;

$$\frac{dm}{dt} = 0 \tag{16}$$

Momentum ($m\boldsymbol{u}$), on the other hand, can decrease or increase, depending on the sum of forces ($\boldsymbol{f}$) acting on the fluid (Newtons 2nd law);

$$\frac{d(m\boldsymbol{u})}{dt} = \sum \boldsymbol{f} \tag{17}$$

Equation 16 and 17 can be rewritten from extensive (Lagrangian point of reference) into intrinsic (Eulerian point of reference). The relation is written on a generic form, where ϕ is any generic vector field;

$$\frac{D\phi}{Dt} = \frac{\partial \phi}{\partial t} + \boldsymbol{u} \cdot \nabla \phi \tag{18}$$

and Φ is the corresponding extensive property of ϕ, expressed as:

$$\Phi = \int_{m.v} \rho\phi dV \tag{19}$$

Since conservation of quantities applies to extensive properties, the Reynolds transport theorem needs to be used to extend it to a control volume [74];

$$\frac{d\Phi}{dt} = \frac{\partial}{\partial t}\int_{c.v}(\rho\phi) + \int_{c.s}\rho\phi(\boldsymbol{u}\cdot\boldsymbol{n})dS \tag{20}$$

3.1.2 Mass conservation

From equation 19 $\phi = \frac{\partial\Phi}{\partial m}$. When looking at the conservation of mass ($\Phi = m$), then $\phi = 1$. If there are no chemical reactions and that the control volume is constant, the mass in the material volume is constant ($\frac{d\Phi}{dt} = 0$). By applying the Gauss' divergence theorem on the surface theorem, it is transformed into a volume integral [74], resulting in the continuity equation on differential coordinate-free form:

$$\frac{\partial\rho}{\partial t} + \nabla\cdot(\rho\boldsymbol{u}) = 0 \tag{21}$$

3.1.3 Momentum conservation

When looking at the conservation of momentum ($\Phi = m\boldsymbol{u}$), then $\phi = \boldsymbol{u}$ and combining equation 17 with 20. The conservation of momentum is written as;

$$\frac{\partial}{\partial t}\int_{c.v}(\rho\phi) + \int_{c.s}\rho\phi(\boldsymbol{u}\cdot\boldsymbol{n})dS = \sum\boldsymbol{f} \tag{22}$$

By assuming the fluid is a Newtonian, incompressible and constant dynamic viscosity (μ), equation 22 can be re-written into the famous incompressible Navier-Stokes equation:

$$\rho\frac{\partial\boldsymbol{u}}{\partial t} + \rho(\boldsymbol{u}\cdot\nabla)\boldsymbol{u} = \rho\boldsymbol{f}_v - \nabla p + \mu\nabla^2\boldsymbol{u} \tag{23}$$

Where $\boldsymbol{f}_v$ is the body force working on the fluid and p is pressure.

3.1.4 Conservation of scalars

The integral form of the equation describing conservation of a scalar quantity is analogous to the momentum conservation equation, equation 22:

$$\frac{\partial}{\partial t}\int_{c.v}(\rho\phi) + \int_{c.s}\rho\phi(\boldsymbol{u}\cdot\boldsymbol{n})dS = \sum f_\phi \tag{24}$$

The scalar ϕ can be temperature (energy equation) or concentration (specie equation), f_ϕ represents source terms and other transport mechanisms for ϕ, other than convection.

Other means of scalar transport is diffusion, and it is usually described by a gradient approximation, e.g. Fourier's law or Fick's law for heat diffusion and mass diffusion, respectively. The diffusion term (f_ϕ^d)can written as [74]:

$$f_\phi^d = \int_{c.s} \Gamma \nabla \phi \cdot \boldsymbol{n} dS \tag{25}$$

Where Γ is the diffusivity of ϕ. The integral form of the generic scalar conservation equation can be written as:

$$\frac{\partial}{\partial t} \int_{c.v} (\rho\phi) + \int_{c.s} \rho\phi(\boldsymbol{u} \cdot \boldsymbol{n}) dS = \int_{c.s} \Gamma \nabla \phi \cdot \boldsymbol{n} dS + \int_{c.v} q_\phi dV \tag{26}$$

3.2 Multiphase

3.2.1 Eulerian-Lagrangian method

In the Eulerian-Lagrangian approach, the dispersed phase is treated as small rigid spheres, i.e. neglecting internal flow and deformation, considered as point centers of mass in space. The translation motion of the particles is described by the Lagrangian form of Newtons second law;

$$\frac{d}{dt}(m_p \boldsymbol{u}_p) = \boldsymbol{f}_{hp} + \boldsymbol{f}_p + \boldsymbol{f}_E + \boldsymbol{f}_G + \boldsymbol{f}_D + \boldsymbol{f}_V + \boldsymbol{f}_L + \boldsymbol{f}_B \tag{27}$$

where m_p is the mass of the particle ($m_p = \rho_p V_p$), $\boldsymbol{u}_p$ is the velocity vector of the particle. The RHS of equation 27 denote surface and body forces due to ; hydrostatic pressure ($\boldsymbol{f}_h p$), gravity ($\boldsymbol{f}_G$), external fields, i.e magnetic field, ($\boldsymbol{f}_E$), drag ($\boldsymbol{f}_D$), virtual mass ($\boldsymbol{f}_V$), transversal lift ($\boldsymbol{f}_L$) and Basset history force ($\boldsymbol{f}_B$). The particle's movement is calculated from the definition of the translational velocity of the center of mass [59]:

$$\frac{d\boldsymbol{r}_p(t)}{dt} = \boldsymbol{u}_p(t, \boldsymbol{r}_p(t)) \tag{28}$$

- **One-way coupled** system assumes the volume of the dispersed phase is so small that the forces acting on the continuous phase from to the dispersed phase is negligible, i.e. the (local) flow of the continuous phase affects the dispersed phase, while the opposite is assumed negligible. Commonly used if $\alpha_{\text{dispersed}} < 10^{-6}$.
- **two-way coupled** systems assumes does not neglect the forces from the dispersed phase on the continuous phase. Can be used if $\alpha_{\text{dispersed}} < 10^{-3}$.
- **four-way coupled** systems also take into consideration the particle-particle interaction and any turbulence generated due to the presence of the dispersed phase.

In a one-way coupled system, the Eulerian velocity is calculated for the continuous phase, similarly to the calculation of a single-phase system. The particle's movement is calculated afterward, independently of other particles. In a two-way coupled system, the fully time-dependent equations for the dispersed and continuous phase are solved iteratively, taking into consideration the interaction between the different phases. In a four-way coupled system there is an interaction between phases (as in a two-way coupled system), but in addition, there is a mutual interaction between the particles in the dispersed phase. In the case of a four-way coupled system, the computational cost gets high due to its complexity, in these cases, the Eulerian-Eulerian method can be a better solution. In Eulerian-Lagrangian models, the disperse phase is studied by assessing either the motion of individual particles or the motion of particle agglomerates (parcels) [75].

3.2.2 Eulerian-Eulerian method

The Eulerian-Eulerian, also referred to as the two-fluid model or the multi-fluid model, treats both phases as continuous phases. The movement of the phases is calculated by using the partial differential Navier-Stokes equation, typically involving multiple averaged Navier-Stokes equations.

The Eulerian-Eulerian approach is more suitable in cases of dense two-phase flow, i.e. significant volume fractions of both phases. In these cases, the assumption that both phases are continuous becomes more realistic [76]. The movement of the phases is calculated by the partial differential Navier-Stokes equation, typical involving sets of averaged Navier-Stokes equations. The continuity equation, equation 21, is rewritten as;

$$\frac{\partial}{\partial t}(\alpha_k \rho_k) + \nabla(\alpha_k \rho_k \boldsymbol{u}_k) = \dot{S}_k \tag{29}$$

where α_k is the volume fraction of phase k, ρ_k is the density of phase k, $\boldsymbol{u}$ is the velocity of phase k and S_k is the phase transfer source term. For a two-phase simulation consisting of a solid and gas phase, the continuity equations would be;

$$\frac{\partial}{\partial t}(\alpha_g \rho_g) + \nabla(\alpha_g \rho_g \boldsymbol{u}_g) = \dot{S}_g \tag{30}$$

$$\frac{\partial}{\partial t}(\alpha_s \rho_s) + \nabla(\alpha_s \rho_s \boldsymbol{u}_s) = \dot{S}_s \tag{31}$$

where $\alpha_s + \alpha_g = 1$, $\dot{S}_g$ and $\dot{S}_s$ is the change due to phase transfer.

The momentum equation for the fluid and solid can be written as;

$$\frac{\partial}{\partial t}(\alpha_f \rho_f \boldsymbol{u_f}) + \nabla \cdot (\alpha_f \rho_f \boldsymbol{u}_f \boldsymbol{u}_f) = -\alpha_f \nabla p + \alpha_f \rho_f \boldsymbol{g} + \nabla \cdot \tau_f + I_f + M_f \tag{32}$$

$$\frac{\partial}{\partial t}(\alpha_s \rho_s \boldsymbol{u_s}) + \nabla \cdot (\alpha_s \rho_s \boldsymbol{u}_s \boldsymbol{u}_s) = -\alpha_s \nabla p - \Delta p_s + \alpha_s \rho_s \boldsymbol{g} + \nabla \cdot \tau_s + I_s + M_s \tag{33}$$

where I_k is the inter-phase momentum transfer. The inter-phase momentum transfer term, is to ensure the conservation of momentum. In the case of mass transfer between phases, an additional

term is added to the momentum equation. The term (M_k) takes into account the momentum entering and leaving the phase during mass transfer and between phases and is defined as;

$$M_k = \dot{m}_{k,entering}\boldsymbol{u}_j - \dot{m}_{k,leaving}\boldsymbol{u}_k \tag{34}$$

where $\dot{m}$ is the mass transfer rate, and is equal to $\dot{S}_k$. τ is the stress tensor, and can for the fluid be calculated by;

$$\tau_f = \alpha_f\mu_f(\nabla\boldsymbol{u}_f + \nabla\boldsymbol{u}_f^T) - \frac{2}{3}\alpha_f\mu_f\nabla\cdot\boldsymbol{u}_f\bar{\boldsymbol{I}}_{\boldsymbol{f}} \tag{35}$$

where $\bar{\boldsymbol{I}}$ is the unit tensor, λ_f is the bulk viscosity and μ_f is the fluid dynamic viscosity. The solid stress tensor can be calculated as [77];

$$\tau_s = \alpha_s\mu_s(\nabla\boldsymbol{u}_s + \nabla\boldsymbol{u}_s^T) + \alpha_s(\lambda_s - \frac{2}{3}\mu_s)\nabla\cdot\boldsymbol{u}_s\bar{\boldsymbol{I}}_{\boldsymbol{s}} \tag{36}$$

The solid phase is modeled as a fluid in the Eulerian method, models describing the particle-particle interactions are added, usually as a source/sink term in the momentum equation, ∇p_s, or included in the viscosity [62].

Hence, the Eulerian–Eulerian models neglect the discrete nature of the disperse phase. To characterize the properties of the material this approach uses averaging techniques [75].

3.3 Inter-phase Momentum Transfer

In a two phase system, in the Eulerian-Eulerian frame of reference, the momentum transfer between the the two phases are coupled together by the inter-phase momentum transfer term, I_k in equation 32 and 33, with oposite signs;

$$I_f = -I_s \tag{37}$$

Some of the possible inter-phase components can be but are not limited to drag, lift and turbulence dispersion [78]. In the case of particles with a discrete size distribution, d_p is replaced by the Sauter mean diameter $\mathrm{d_{p,SM}}$;

$$d_{p,SM} = \frac{1}{\sum\frac{x_i}{d_{p,i}}}$$

3.3.1 Drag force

The drag force is the frictional force working on the particle, generated by the gas flow. The magnitude of the drag force increases with increasing superficial velocity ($\mathrm{u_s}$) between the particle and the gas. The drag equation, equation 38, is a general formula to calculate the drag force on a particle.

$$F_D = \frac{1}{2}\rho_g C_D A_{cs} |u_r| u_r \tag{38}$$

In multiphase flow the drag force acting on the particles in a volume can be expressed as [79];

$$F_D = \frac{3}{4}\rho_f \alpha_f \alpha_p C_D \frac{|\boldsymbol{u}_r|\boldsymbol{u}_r}{d_p} \tag{39}$$

Where ρ is the density of the gas, C_D is the drag coefficient, A_{cs} is the exposed cross-section area of the particle. $\boldsymbol{u_r}$ is the relative velocity, i,e the velocity difference between the particle and the gas.

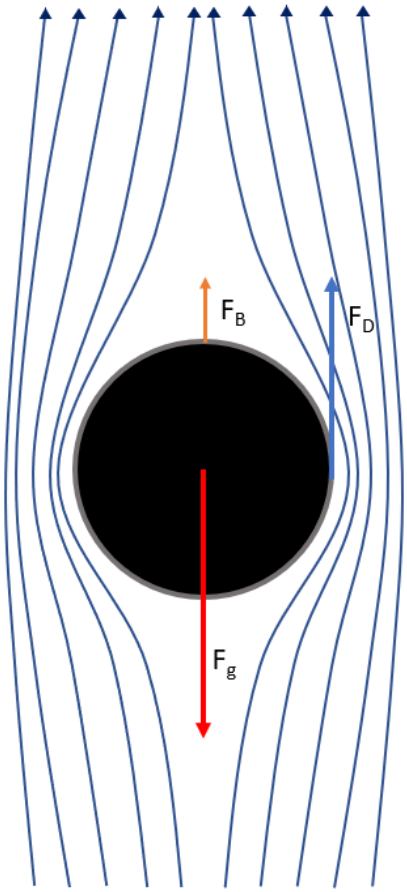


Figure 3.2: Shows the drag-lift and gravity force on a particle with a fluid flow upwards

In multiphase flow, the drag force has shown to be the critical closure model to accurately predict the gas-solid multiphase flows. Research has proposed multiple drag models based on the flow conditions [80]. Upadhyay et al., [80] gives an overview and assessment of the most common drag models in E-E CFD simulations. The two models that predicted the drag force most accurate are presented here;

Syamlal-O'Brian drag model;

$$F_D^{sf} = \frac{3}{4} C_D \frac{\alpha_s \alpha_g \rho_g}{V_{r,s}^2 d_p} \left(\frac{Re_d}{V_{r,s}} \right) |\boldsymbol{u}_s - \boldsymbol{u}_s| \tag{40}$$

where C_D is expressed as;

$$C_D = \left(0.63 + \frac{4.8}{\sqrt{\frac{Re_d}{V_{r,s}}}} \right)^2$$

$$V_{r,s} = 0.5(A - 0.06Re_D + \sqrt{(0.06Re_d)^2 + 0.12Re_d(2B - A) + A^2}$$

$$\begin{aligned} A &= \alpha_g^{4.14} \\ B &= 0.8\alpha_g^{1.28} && \text{for } \alpha_g \leq 0.85 \\ B &= 0.8\alpha_g^{2.65} && \text{for } \alpha_g > 0.85 \end{aligned}$$

and the Reynolds number, Re_d is expressed as equation 41 for both Syamlal-O'Brian drag model and Gidaspow's.

$$Re_d = \frac{\rho_g d_p |\boldsymbol{u}_s - \boldsymbol{u_g}|}{\mu_g} \tag{41}$$

Gidaspow's drag model is a combination of Wen-Yu and Erguns drag models;

$$\begin{aligned} F_D^{sf} &= \frac{3}{4} C_D \frac{\alpha_s \alpha_g \rho_g |\boldsymbol{u}_s - \boldsymbol{u_g}|}{d_p} \alpha_g^{-2.56} && \text{for } \alpha_g > 0.8 \\ F_D^{sf} &= 150 \frac{\alpha_s^2 \mu_g}{\alpha_g d_p^2} + 1.75 \frac{\alpha_s \rho_g |\boldsymbol{u}_s - \boldsymbol{u}_g|}{d_p} && \text{for } \alpha_g \leq 0.8 \end{aligned} \tag{42}$$

where the drag coefficient, C_D is expressed as:

$$\begin{aligned} C_D &= \frac{24}{\alpha_g Re_d} (1 + 0.15 \alpha_g Re_d^{0.0687}) && Re_d < 1000 \\ C_D &= 0.44 && Re_d \geq 1000 \end{aligned}$$

3.3.2 Virtual mass force

Drag and virtual mass forces are the most important components of the interfacial momentum transfer. The force from the virtual mass effect is small compared to the drag force, but with strong relative acceleration, the correct modeling of the virtual mass force is of fundamental importance to achieve accurate predictions of the flow variables, including pressure and velocity fields [81].

Virtual mass (also referred to as added mass) is a phenomenon that determines the necessary work done by the particle moving in the fluid to change the kinetic energy associated with the motion of the fluid. When a particle moves through a quiescent or relatively slowly moving liquid, some liquid mass is carried by the particle along with it. This portion of the liquid mass is supposed to attain the particle velocity. This results in a virtual increase in the particle mass, originating the concept of virtual mass in the mixture. The virtual mass force for an accelerating particle in a continuous fluid phase can be expressed as [82]:

$$F_{vm} = \mathscr{C}\rho_f V_p \boldsymbol{a}_{VM} \tag{43}$$

The formulation of virtual mass introduces the virtual mass coefficient, $\mathscr{C}$, which describes the fraction of displaced fluid that contributes to the effective mass of the particle. A common approach is to set the virtual mass coefficient as a constant, $\mathscr{C} = 0.5$, which is the added mass on a spherical particle [83]. $\boldsymbol{a}_{VM}$ is the virtual mass acceleration written as [82];

$$a_{VM} = \frac{\partial \boldsymbol{u}_R}{\partial t} + (\boldsymbol{u}_p \cdot \nabla)\boldsymbol{u}_p - (\boldsymbol{u}_f \cdot \nabla)\boldsymbol{u}_f \tag{44}$$

3.3.3 Other forces

Thermophoresis is defined as the migration of a microscale particle in a fluid as a response to a temperature gradient in the fluid [84].

Basset Force, also referred to as history force and arises due to acceleration between the continuous and dispersed phase and the delayed development of the boundary layer near the interfacial surface [85].

Calculating the basset force on a particle is time and memory consuming, as every earlier time step and integral must be evaluated for the entire simulation [86].

The Magnus force produces a velocity component that is perpendicular to the net force experienced by the particle [87].

The saffman's lift force is a lift force generated due to local velocity gradient in the fluid flow, i.e. shear flow [88].

3.4 Energy Equation

The energy equation are solved for the system to ensure that energy is conserved. The energy equation can be expressed as [71]:

$$\frac{\partial \rho_k C_{p,k} \alpha_k T_k}{\partial t} + \nabla \cdot \rho_k c_{p_k} \alpha_k \boldsymbol{u}_k T_k = \alpha_k k_k \nabla T + Q_I + Q_{r,k} \tag{45}$$

were $C_{p,k}$ is the specific heat capacity for phase k, k_k is the thermal diffusivity, Q_I is the heat source term for heat transfer between phases, and $Q_{r,k}$ is the source term for the energy generated inside the phase. This thesis will not go more into the energy equation or heat transfer as the system is assumed isotherm. More information can be found in the book *Fundamentals of Multiphase Heat Transfer and Flow* [89].

3.5 Specie Equation

The scalar equation for the individual species in their respective phases are calculated by [71];

$$\frac{\partial \alpha_k y_{i,k}}{\partial t} + \nabla \alpha_k \boldsymbol{u}_k y_{i,k} = \nabla \cdot D_{i,k} \nabla y_{i,k} + \dot{M}_{i,k} + \dot{R}_{i,k} \tag{46}$$

where $\dot{R}_{i,k}$ is the source term due to chemical reactions, $\dot{M}_{i,k}$ is the mass transfer between phases. The adsorption rate can be calculated by different models, the most common way of describing an adsorption reaction is with the Langmuir isotherm. There are proposed other adsorption models, Plazinski et al. [90] gives a good overview of these.

3.5.1 Langmuir kinetics Isotherm

Suppose that the adsorption reaction reaches an equilibrium, where a fraction, Γ, of the surface has been covered with the adsorbent. θ is defined as the fraction of current surface coverage of the maximum coverage Γ.

The Langmuir isotherm states that the rate of adsorption will be proportional to the concentration of the adsorbate and the fraction of the surface that is bare, as the reaction only can take place where the molecules are in contact with the bare surface [91];

The Langmuir kinetics [92] is derived from the adsorption process being a reversible process;

$$A_(g) + O_(s) = [k_1][k_2]A(s) \tag{rx.5}$$

Then by assuming that the forward adsorption rate is first order with respect to [A] and the fraction of the surface covered, (1-$\theta(t)$), the adsorption and desorption rates are [92];

$$r_a = k_a[A_g](1 - \theta(t)) \tag{47}$$

$$r_d = k_d\theta(t) \tag{48}$$

where k_a and k_d is the rate constant for the forward and backward reaction, [A] is the concentration of the adsorbent, and $\theta(t)$ is the fraction of reacted surface sites and is a function of time. The combined adsorption and desorption rate is then;

$$\frac{d[A_s]}{dt} = r_a - r_d = k_a[A_g](1 - \theta(t)) - k_d\theta(t) \tag{49}$$

The adsorption of HF on alumina is considered an irreversible reaction, i.e. k_d=0. This simplifies equation 49 into only the forward reaction term, equation 47:

$$\dot{R}_{i,k} = k_i[A_g](1 - \theta(t)) \tag{50}$$

The adsorption capacity of alumina has proven difficult to determine. One approximation to the adsorption capacity is to use the fluoride loading at the breakthrough point, figure 2.6. This fluoride loading is used as a measurement of adsorption capacity and is defined as the kinetically available adsorption capacity. This assumption is based on the decrease in surface area of fluorinated alumina in figure 2.6. The surface area decreases linearly towards the breakthrough point until the surface area decreases significantly when the reaction time is close to the time of breakthrough. This sudden decrease in the surface area could be a sign of the adsorption sites at the exterior surface are spent. This, in turn, increasing the amount of HF that diffuses into the pores, increasing the pore blocking rate significantly, and reduce the surface area rapidly as pores are blocked. With this assumption the kinetically available surface cite can be expressed as;

$$\theta(t) \simeq \frac{C_{ads}(t)}{C_{BK}} \tag{51}$$

where $C_{ads}(t)$ is the concentration of adsorbate in the adsorbent at time t, and C_{BK} is the concentration of adsorbate in the adsorbent at breakthrough.

3.5.2 Reaction rate constant

The reaction rate of a chemical reaction is affected by temperature. The most commonly used relation between reaction rate and temperature is the Arrhenius equation, stating that the rate constant k is the product of a pre-exponential frequency factor A, and an exponential term E_a [93];

$$k = Ae^{\frac{E_a}{RT}} \tag{52}$$

where R is the gas constant and E_a is the activation energy.

For reactions involving solids, the reaction rate is directly proportional to the total area between the reactants and the solids [94]. Then equation 52 can be rewritten to include the surface area dependency;

$$k = Ae^{-\frac{E_a}{RT}} A_s \tag{53}$$

the reaction rate can then be written as;

$$\dot{R} = [A_g]Ae^{-\frac{E_a}{RT}} A_s(1 - \theta(t)) \tag{54}$$

From equation 54 the $(1 - \theta(t))$ term can be implemented in the reaction rate constant calculations. Then equation 53 can be extended to include the adsorption capacity term

$$k = Ae^{-\frac{E_a}{RT}} A_s(1 - \theta(t)) \tag{55}$$

The chemical reaction is one aspect of the adsorption process, another is the mass transfer of a specie in the bulk gas to the surface of a solid.

3.5.3 Mass transfer

In reality, when fluid flows around a particle or another phase where the velocities are different, a boundary layer is formed between the interfaces, figure 3.3 shows a schematic of the boundary layer formed around a particle. The mass transfer boundary layer thickness is defined as the distance from the particle surface to where the concentration is 99% of the bulk concentration. Nearly all resistance towards the mass transfer is inside the mass transfer layer [95].

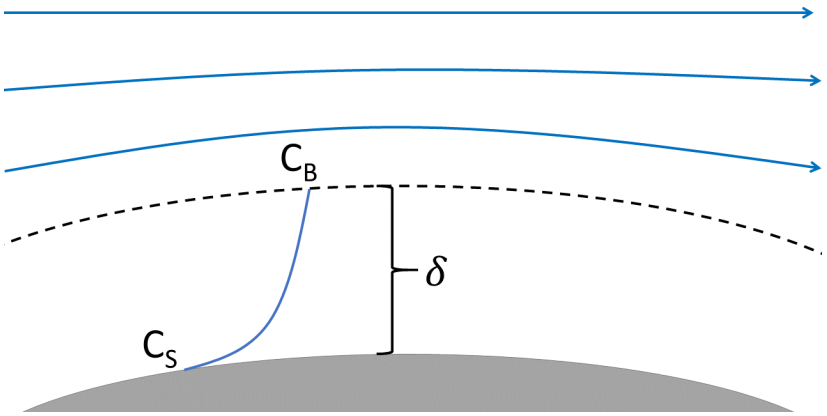


Figure 3.3: Shows the boundary layer formed around a particle

3.5.4 Phase transfer

The change in the phase momentum in equation 32 and 33 is due to phase transfer, and is in turn dependent on the specie transfer term $M_{i,k}$, in the specie equation. The specie transfer term in equation 46, is describing the adsorption of the adsorbate, which in turn is controlled by the chemical reaction rate. Because of this, the specie transfer rate is equal to the chemical reaction rate;

$$|\dot{M}| = |\dot{R}| \tag{56}$$

The phase transfer term should be reflected in the source term of the phase conservation equations, equation 30 and 31 and the momentum equations 32 and 33. The sum of all phase transfers should be equal to the source term, S in phase k.

$$\dot{S}_k = \sum_{i=1}^{N} \dot{M}_{i,k} \tag{57}$$

3.6 Summery of Solver

The model will simulate the adsorption of hydrogen fluoride on alumina particles in the Eulerian-Eulerian frame of reference. To describe the adsorption reaction the hydrogen fluoride gas reacts on the alumina surface into a temporary gas, this temporary gas is then transferred to the solid particles with a rate that is equal to the reaction rate generating the temporary gas. The system is assumed isotherm, the flow is assumed laminar and there is no resistance to phase transfer between the phases, i.e the chemical reaction rate is the rate-determining step.

This model is essentially described by the conservation of two phases with the phase momentum conservation equations 32 and 33. The particle-particle interaction in the solid phase is not considered. The species in their respective phases are described with equation 46 where the reaction term is described with equation 54 and the phase transfer term by equation 56. The reaction term, $R_{i,k}$ is equal to the specie transfer, $M_{i,k}$, and the sign of these are determined by the direction of the reaction and phase transfer.

The system is described with the following PDEs;

$$\frac{\partial}{\partial t}(\alpha_g \rho_g) + \nabla(\alpha_g \rho_g \boldsymbol{u}_g) = -\dot{S}_g$$

$$\frac{\partial}{\partial t}(\alpha_s \rho_s) + \nabla(\alpha_s \rho_s \boldsymbol{u}_s) = \dot{S}_s$$

$$\frac{\partial}{\partial t}(\alpha_f \rho_f \boldsymbol{u_f}) + \nabla \cdot (\alpha_f \rho_f \boldsymbol{u}_f \boldsymbol{u}_f) = -\alpha_f \nabla p + \alpha_f \rho_f \boldsymbol{g} + \nabla \cdot \tau_f - F_D - F_{VM} + \dot{S}\boldsymbol{u_g}$$

$$\frac{\partial}{\partial t}(\alpha_s \rho_s \boldsymbol{u_s}) + \nabla \cdot (\alpha_s \rho_s \boldsymbol{u}_s \boldsymbol{u}_s) = -\alpha_s \nabla p - \alpha_s \rho_s \boldsymbol{g} + \nabla \cdot \tau_s + F_D + F_{VM} + \dot{S}\boldsymbol{u_g}$$

$$\frac{\partial \alpha_k y_{i,k}}{\partial t} + \nabla \alpha_k \boldsymbol{u}_k y_{i,k} = \nabla \cdot D_{i,k} \nabla y_{i,k} + \dot{M}_{i,k} + \dot{R}_{i,k}$$

where the source term in the continuity equation is described as the sum of all mass transfer due to phase transfers:

$$\dot{S}_k = \sum_{i=1}^{N} \dot{M}_{i,k}$$

The frag force, F_D, is described by Gidaspow's drag model:

$$F_D = \frac{3}{4} C_D \frac{\alpha_s \alpha_g \rho_g |\boldsymbol{u}_s - \boldsymbol{u_g}|}{d_p} \alpha_g^{-2.56} \qquad \text{for } \alpha_g > 0.8$$

$$F_D = 150 \frac{\alpha_s (1 - \alpha_g) \mu_g}{\alpha_g d_p^2} + 1.75 \frac{\alpha_s \rho_g |\boldsymbol{u}_s - \boldsymbol{u}_g|}{d_p} \qquad \text{for } \alpha_g \leq 0.8$$

The virtual mass force is described with:

$$F_{vm} = \mathscr{C}\rho_f V_p \left(\frac{\partial \boldsymbol{u}_R}{\partial t} + (\boldsymbol{u}_p \cdot \nabla)\boldsymbol{u}_p - (\boldsymbol{u}_f \cdot \nabla)\boldsymbol{u}_f \right)$$

The specie reaction term are described by:

$$\dot{Ri}, k = [Y_i, k] A e^{-\frac{E_a}{RT}} A_s (1 - \theta(t))$$

and the phase transfer term is equal to the reaction term:

$$|\dot{M}| = |\dot{R}|$$

The PDEs are solved numerically in the CFD program OpenFOAM.

4 Numerical Framework

4.1 Computational Fluid Dynamics

Fluid flow is caused by external forces applied on the fluid, such as pressure, gravity and shear forces. The flow can be described by PDEs, which in most cases are impossible to solve analytically. Computational fluid dynamics programs use different discretization methods to approximates the PDEs into a system of algebraic equations, when solved for small domains in space and time, gives an approximate numerical solution for the fluid flow. Similar to the accuracy of experimental work depends on the equipment, the accuracy of numerical solutions depends on the quality of the discretization used. It is important to bear in mind that numerical results are always an approximation. There are different reasons for the difference between computed results and reality, as errors are introduced from each step to calculate the numerical solution [96].
The basis for any CFD simulation is a mathematical model, i.e, a set of PDEs describing the flow and boundary conditions. A simulation can never be better than the mathematical model describing the system. There are done approximations in the discretization process, i.e. the process that transforms PDEs into sets of algebraic equations. The algebraic equations are solved numerically, i.e. iterative methods are used. Unless the calculations run for a long time the simulations are approximations of the real solution. Different CFD software uses different methods to discretize the PDE's, most CFD programs are based on the finite volume method (FVM) and Finite Element Method (FEM). Some of the most famous CFD software are;

Table 4.1: CFD software and their discretization method

Software	Discretization method
Ansys Fluent	FVM
Ansys CFX	FVM
ADINA	FEM (FVM)
COMSOL	FEM
OpenFOAM	FVM

All CFD solution strategies consists of three separate stages;

- Pre-processing.
- Solving.
- Post-processing.

In this thesis, OpenFOAM is utilized to solve the PDEs.

4.1.1 Discretization

An analytic solution of the PDEs is almost impossible, thus the solution must be approximated. The translation of PDEs into a set of algebraic equations is referred to as discretization or the discretization process. OpenFOAM uses the FVM to discretize the PDEs.

There are in short three steps to the discretization of the PDEs with the finite volume method;

- Domain Discretization
- Equation discretization
- Solving the algebraic equation

The complete method of discretization methods is out of the scope for this thesis, but a simple explanation of the process is given. Domain discretization results in a grid system of non-overlapping sub-domains, referred to as cells or control volumes, that fill the computational domain. This process is referred to as meshing or mesh generation. The cells can have different shapes, and pyramids, squares, and hexahedrons are used, but the cells must be convex and the faces must be planar [97]. The algebraic equations from the discretization of the PDEs are described for each cell, where the solution is expressed as an element field. OpenFOAM's finite volume method uses a co-located grid, i.e the fluid dynamic properties are stored in the control volume centroid [98]. The complete mesh is made out of cells, faces (interfaces) and nodes (cell centers), as sketched in figure 4.1.

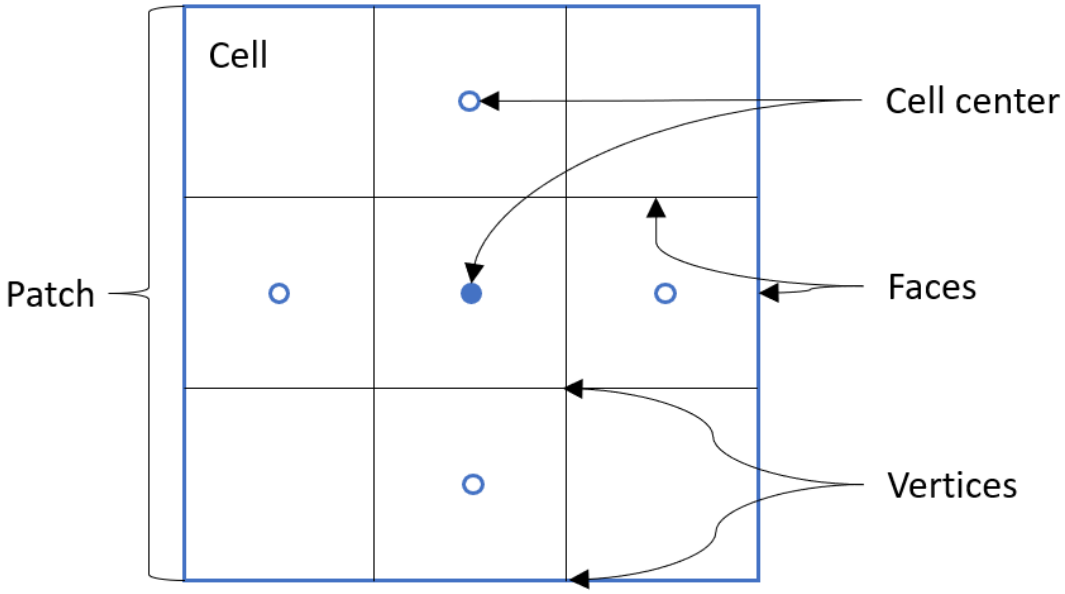


Figure 4.1: Shows some of the component in a mesh

Equation discretization is the transformation from PDEs into sets of algebraic equations. These algebraic equations are assembled into a global matrix and vectors that can be expressed in the form $\boldsymbol{A}[\phi] = \boldsymbol{b}$, where the unknown variable ϕ is defined in each cell centroid and at the boundary of the computational domain as a boundary condition [99].

The finite volume method discretizes the PDEs by integrating them over the control volume, i.e cell volume. The values of the faces of the cells are obtained by interpolation between neighboring cell centers. The value of ϕ in a cell only affects its neighboring cells. It is expected that the solution of the discretized equation approaches the exact solution of the corresponding PDE as the distance

between cells gets smaller and the change in ϕ becomes smaller, resulting in the details of the profile assumption becoming smaller

Solving the discretized equations implies solving a set of discrete algebraic equations. The technique to solve the algebraic equations is independent of the discretization method. The iterative method is a guess-and-improve procedure, to finalize an answer that satisfies the tolerance chosen. How the solver solves the numerical iteration is decided by the chosen numerical scheme. Each cell is updated with a new value of ϕ, resulting in an update of the value of ϕ in the neighboring cells, until the difference in updates is smaller than the specified tolerance. The tolerance mesh and time step is important factors when reducing the numerical errors in the simulation.

4.2 Inaccuracy in CFD Models

An apparently good CFD simulation can be completely incorrect. In the best case, the incorrect simulation might have only wasted time and money and not lead to catastrophic failures. Any reasonable estimate of numerical errors is better than none, and because numerical solutions are approximate solutions, their accuracy should always be questioned [100]. In the context of confidence and trust in CFD modeling, the following definitions of error and uncertainty have been widely accepted [101]: Numerical errors compromises of discretization error, iterative convergence error, and round-off errors. Coding error is errors that are generated from bugs, i.e. coding error, in the source code. User errors are generated from the incorrect use of the code or over-idealizing the system.

4.2.1 Numerical errors

Discretization errors comes from the transformation of PDEs into a set of algebraic equations. The algebraic equation represents the PDEs, but are not exactly the same, and the differences adds up [102]. If the PDEs results in result v, than algebraic equation will yield a result v', then;

$$v = v' + e'$$

where e' represents the dizcretization error. In general as time steps and cell size decreases, so does the dizcretization error. Solving the set of algebraic equations requires an iterative process, e.g the pressure and velocity coupling. For each iteration the solution gets closer to the real solution of the algebraic equations, increasing the number of iteration should increase the accuracy of the answer, but increasing the number of iteration will prolong the calculation time. The round-off error is generated from the computes limitation to store an infinite number of decimals. Every step that is made in a calculation is done with a finite number of significant decimals, each calculation, therefore, introduces a round-off error. The round-off error is usually small and in a well-written code, it can be reduced, but never completely avoided. The addition of the round off error could be written as

$$v' = V + \epsilon$$

where v' is the result from solving the algebraic equation, V is the result computed and ϵ is the round-off error. Commonly in CFD literature the solution v is called correct solution, v' the exact solution and V the computed solution [102].

4.2.2 Coding error

With increasingly complex models, programming mistakes can occur in the code. A successful comparison of real flow to simulated flow, is not a guarantee fault-free CFD code, as the fault might not affect this specific simulation [102]. How much the bug in the system affects the simulation is completely dependent on the simulation.

4.2.3 User error

The number of people using CFD software is increasing [102]. If a user is new to CFD and numerical mathematics he could be a source of errors by using the code wrong. Not only unskilled users might affect the results, Peng et al. [103] did a study where the results from 22 different teams and compared their results. They found that results differed due to the use of different codes, e.g turbulence models, numerical schemes, but also when the same model was used. Possible errors in decisions based on users' experience may have caused the observed significant difference.

While the different CFD software utilizes similar methods to solve the PDE's, the user interface and price of the different software are quite different.

4.3 OpenFOAM

Open source Field Operation and Manipulation, short OpenFOAM, is an open source code written in C++, based on the finite volume method, capable of solving industrial type problems [104]. The goal for OpenFOAM and its developers is well summarised in a citation from the OpenFOAM Direct [105]:

> *After many years, OpenFOAM still is and always will be free to use, share and develop. We hope it is useful for your work in CFD and you enjoy the freedoms of open source software. That includes its unlimited customisation and free deployment on massively parallel computers. Currently, development of the next version of OpenFOAM and maintenance of the current one are principally undertaken by the team at CFD Direct with contributions from a growing community of open source CFD enthusiasts. Please join the community in supporting the OpenFOAM project.*

OpenFOAM can be used to build custom solvers or use some of the ready solvers to solve problems in continuum mechanics. At the core of OpenFOAM are a set of object classes that allows the programmer to manipulate meshes, geometries and dizcretization techniques. In OpenFOAM the algorithms are written in a natural way. The discretization of the transport equation for a scalar ϕ is given by;

$$\underbrace{\frac{\partial \phi}{\partial t}}_{\text{accumulation}} + \underbrace{(\nabla \cdot \boldsymbol{u})\phi}_{\text{convection}} - \underbrace{D\nabla^2\phi}_{\text{diffusion}} = \pm \underbrace{\dot{S}_\phi}_{\text{Source/Sink term}}$$

is written in OpenFOAM as;

```
{
    fvScalarMatrix TEqn
    (
        fvm::ddt(phi)
      + fvm::div(psi, phi)
      - fvm::laplacian(Dphi, phi)
     ==
        fvOptions(phi)
    );
}
```

Listing 4.1: Script for soliving a simple transport equation in OpenFOAM

The fvm::div operator takes the convective flux, psi, as a coefficient field defined over the faces of the control cell and phi as the variable field defined over the cell centroids, and returns a system of equations including a matrix and a source term that represents the discretization of the convection operator.

While different solvers utilize similar methods of solving the PDEs the file structure of each case, and file structure of each solver, are quite different in OpenFOAM

4.3.1 Case structure in OpenFOAM

Cases are built up in one main directory, i.e. folder, within this directory there are three sub-directories; 0, constant and system, which all are holding multiple header files. An example of a general case setup is shown in figure 4.2.

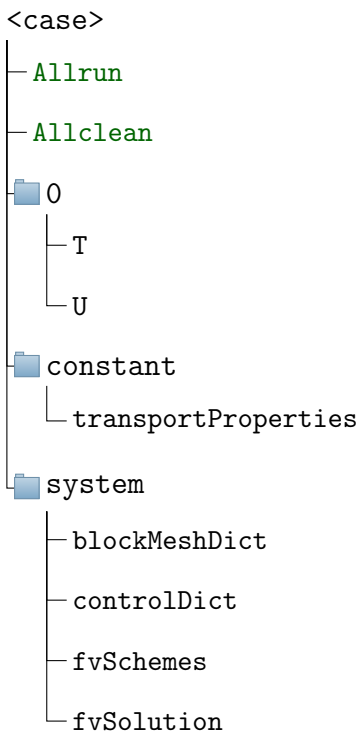


Figure 4.2: A general file structure with the the basic files found in OpenFOAM

The 0-directory contains initial conditions for the different fields, i.e. temperature, species and velocity, as well as the boundary conditions. In OpenFOAM, boundary conditions are defined for every variable at each boundary patch. To get a physical sound system it is necessary for the various variables at one patch to match each other [106]. Each boundary condition has a physical meaning described mathematically by an equation, which in the context of numerical methods must be translated into an algebraic relation.

Physical properties, thermophysical properties, information about the mesh, reaction, or flow regime are specified in the constant directory. The system holds information about the overall system, like parameters associated with the solution procedure, discretization schemes, run-controls, like start time, time-step size and end-criteria. The case set up for multiphaseEulerFoam cases are more complex than the setup in figure 4.2, as the solver needs more information.

4.3.2 PIMPLE

Solving the momentum equation requires numerical techniques for coupling the pressure and momentum quantities. In OpenFOAM this is done by SIMPLE, PISO and PIMPLE algorithms;

- SIMPLE; Semi-Implicit-Method-Of-Pressure-Linked-Equations
  In OpenFOAM this is used for steady-state.
- PISO; Pressure-Implicit-Split-Operator
  Transient calculations are limited by the Courant number < 1.

- PIMPLE; Merged PISO–SIMPLE
  The combined algorithm allows for Courant number > 1.

The Courant number (Co) is a measurement of the rate information is transported in the system and a limiting actor for the performance of numerical schemes. In OpenFOAM it is calculated as;

$$Co = \Delta_t \tau$$

where Δ_t is the time-step size τ is the characteristic time scale based on the local cell flow scale;

$$\tau = \frac{1}{2V} \sum_{faces} |\phi_i| \tag{58}$$

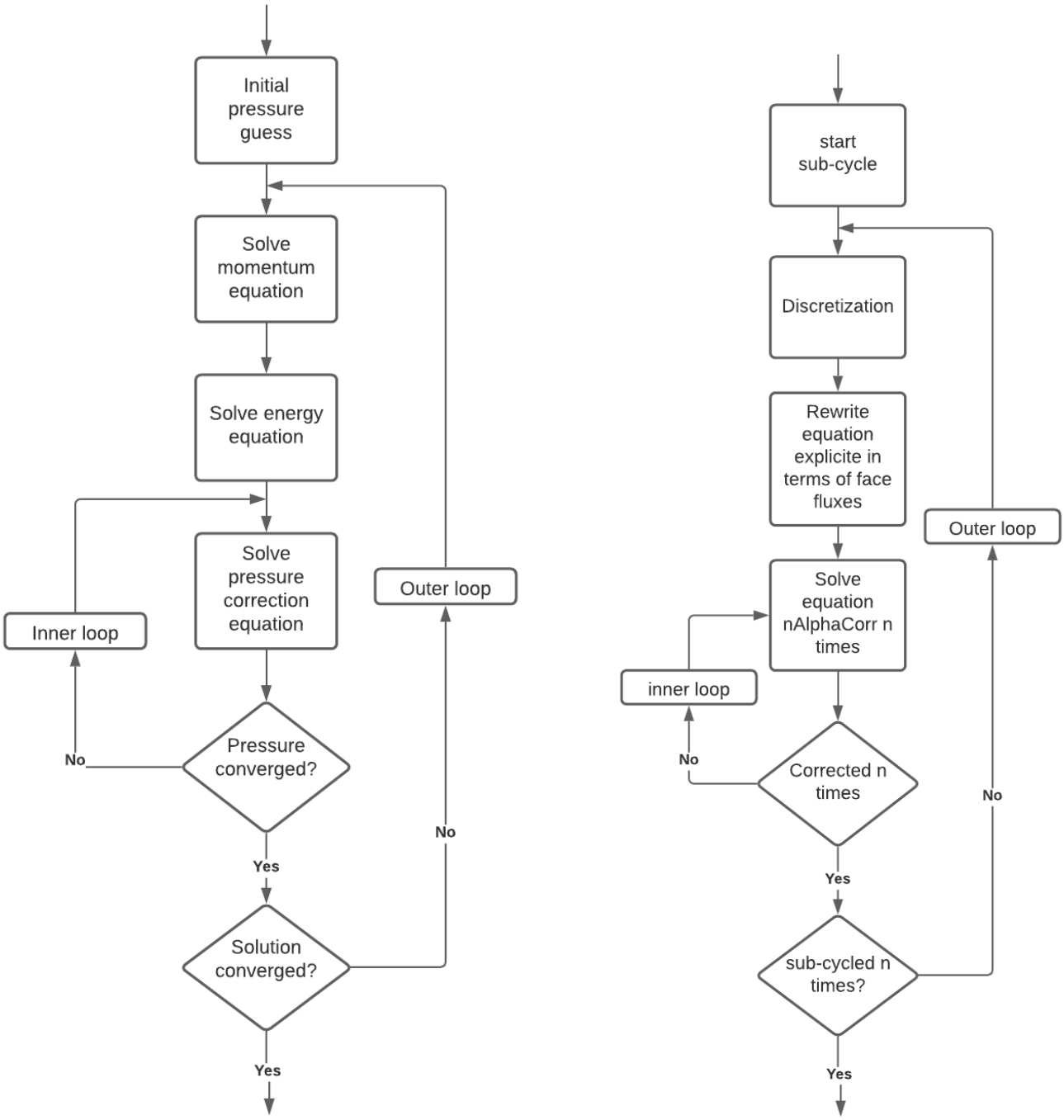
where V is the volume, $\sum_{faces} |\phi_i|$ is the summation of volumetric fluxes over all faces [107].

The principle behind the PIMPLE algorithm is that within one time step, a steady-state solution is attempted reached with under-relaxation. If steady state is achieved or the outer loop has done n corrections, (specified in the fvSolution) the solver goes on to the next time step, schematic of the PIMPLE loop is shown in figure 4.3a. If nOuterCorrectors are 100, and nCorrectors (inner loop) are 2, the PIMPLE algorithm, for one time step, does 100 momentum-pressure corrections and 200 pressure corrections, thus allowing for a bigger time step.

Under-relaxation, or relaxation, are used to increase the stability of the calculations by limiting the rate of change of fields or equations[108], more information can be found in *MATHEMATICS, NUMERICS, DERIVATIONS AND OPENFOAM* [109].

4.3.3 MULES

In multiphase simulations, there is usually used an additional correction loop for the phase fraction fields. The MULES (Multidimensional Universal Limiter with Explicit Solution) algorithm is designed to work as an extra correction for the phase fraction α_k [110]. Another extension is the possibility of setting global extrema for the problem, which is a key feature in multiphase flows, a more in-depth explanation can be found in the PhD-thesis of Damián [111]. Figure 4.3b shows the schematic of the MULES loop, the number of corrector loops are defined in the fvSolution file, by the nAlphaSubCycles parameter.



(a) PIMPLE loop, created from [112]

(b) MULES loop, created from [110]

Figure 4.3: Shows the flow chart of the PIMPLE and MULES loop. For the inner and outer loops the the solution must converge, or have looped over n times many times, specified in fvSolution.

4.4 MultiphaseEulerFOAM

MultiphaseEulerFOAM is a solver capable of solving a system of any number of compressible fluids phases with common pressure and can handle multiple species and in-phase reactions, momentum transfer, heat and mass transfer. For OpenFOAM 8, the developers consolidated the three multiphaseEuler solvers (reactingMultiphaseEulerFoam, twoPhaseEulerFoam and reactingTwoPhaseEulerFoam) into a new and improved solver, multiphaseEulerFoam [113].

"The phase system solution has been enhanced to handle two phases more effectively and all two-phase specific models updated for compatibility so that multiphaseEulerFoam can also replace reactingTwoPhaseEulerFoam. When running multiphaseEulerFoam with only two phases the default behavior is to solve for both phase-fractions but optionally a reference phase can be specified so that only the other phase-fraction is solved, providing better compatibility with the behavior of reactingTwoPhaseEulerFoam." [114]

Each phase is made out of multiple species, each with its own material parameters that are defined in the thermophysicalProperties-files. The properties of the phases are a product of the properties of the species in the phase.

4.4.1 ThermophysicalProperties

In the thermophysicalProperties-folder the thermophysical models concerning energy, heat and physical properties are defined. In the case of two-phase flow, there are commonly two folders, one for each of the phases. A thermophysical model is constructed by the solver in OpenFOAM, as long as the solver uses the thermophysical library. There is one compulsory dictionary entry in thermophysicalProperties, the thermoType, listing 4.2. In this dictionary, the different packages of thermophysical modeling are specified.

```
thermoType
{
    type            heRhoThermo;
    mixture         multiComponentMixture;
    transport       const;
    thermo          hConst;
    equationOfState rhoConst;
    specie          specie;
    energy          sensibleInternalEnergy;
}
```

Listing 4.2: Example entry for the thermoType dictionary

The *type* keyword in thermoType, specifies the underlying thermophysical model. The *mixture* specifies the specie-mixture in the phase. The *transport* keyword specifies which model used to evaluate the dynamic viscosity (μ), thermal conductivity (κ) and thermal diffusion (α^T). Sutherland calculates; μ with the Sutherland coefficients A_s and T_s, equation 59. If the transport is *const*, it means that the dynamic viscosity (μ) is constant and Pr=$\frac{C_p \mu}{\kappa}$, meaning that the μ and Pr must be defined.

$$\mu = \frac{A_s \sqrt{T}}{1 + \frac{T_s}{T}} \tag{59}$$

```
thermodynamics
{
    Tlow            200;
    Thigh           6000;
    Tcommon         1000;
    highCpCoeffs    ( 2.50000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
    0.00000000e+00 -7.4537500e+02 4.37967491e+00 );
    lowCpCoeffs     ( 2.50000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
    0.00000000e+00 -7.4537500e+02 4.37967491e+00 );
}
```

Listing 4.3: Example entry for the thermodynamics

Thermo specifies the thermodynamic model used to calculate the specific heat capacity (C_p). If *thermo* is set to *hconst*, the C_p and H_f is constant and must be defined. Another model is *janaf*, it calculates specific heat capacity from temperature and two sets of coefficients, equation 60, example in listing 4.3. The model is valid between a lower, Tlow, and higher temperature Thigh, and the highCpCoeffs and lowCpCoeffs is used depending on if T > Tcommon. The equation is for calculating the heat capacity is:

$$C_p = R(a_4T^4 + a_3T^3 + a_2^2T + a_1T + a_0) \tag{60}$$

The equation of state decides the model used to calculate the density of the phase. *rhoConst* refers to the density being constant ($\rho = constant$), where as the perfect gas calculates the density, equation 61, as a function of temperature and pressure.

$$\rho = \frac{1}{RT}p \tag{61}$$

The formula used to calculate the gas density in paraview:

$$\rho_g = \frac{p0.001M}{8.3144T} \tag{62}$$

The last specification is specifying what form of energy the model should use; internal energy, enthalpy, or heat of formation.

4.5 MultiphaseEulerFoam.C

The main file in the multiphaseEulerFoam solver is the multiphaseEulerFoam.C [115]. The PIMPLE loop in the solver makes out most of the file and can be seen in listing 4.4 The solver looks up the number of energy corrections, and calculates the Courant number, and adjusts the time step in lines 79-100, before entering the outer loop of the PIMPLE-loop. Inside the PIMPLE-loop the semi-discrete form of the phase continuity equations are solved using the MULES-loop, line 107-109. Then the specie equations are solved for all species, then the velocity-pressure coupling and energy equation are solved inside the inner loop, line 113-124. If there are any turbulence models, it is corrected for at the end of the loop.

```
while (pimple.run(runTime))
    {
        #include "readTimeControls.H"

        int nEnergyCorrectors
        (
            pimple.dict().lookupOrDefault<int>("nEnergyCorrectors", 1)
        );

        if (LTS)
        {
            #include "setRDeltaT.H"
            if (faceMomentum)
            {
                #include "setRDeltaTf.H"
            }
        }
        else
        {
            #include "CourantNo.H"
            #include "setDeltaT.H"
        }

        runTime++;
        Info<< "Time = " << runTime.timeName() << nl << endl;
            // --- Pressure-velocity PIMPLE corrector loop
        while (pimple.loop())
        {
            fluid.solve(rAUs, rAUfs);
            fluid.correct();
            fluid.correctContinuityError();

            #include "YEqns.H"

            if (faceMomentum)
            {
                #include "pUf/UEqns.H"
                #include "EEqns.H"
                #include "pUf/pEqn.H"
            }
            else
            {
                #include "pU/UEqns.H"
                #include "EEqns.H"
                #include "pU/pEqn.H"
            }

            fluid.correctKinematics();

            if (pimple.turbCorr())
            {
                fluid.correctTurbulence();
            }
        }
```

```
132
133         runTime.write();
134
135         Info<< "ExecutionTime = "
136             << runTime.elapsedCpuTime()
137             << " s\n\n" << endl;
138     }
```

Listing 4.4: PIMPLE loop inside the multiphaseEulerSolver

4.6 Species equation in multiphaseEulerFoam

Usually in OpenFOAM, specie concentration, C_i are handled with kgm^{-3} as units. To make the multiphaseEulerFoam solver capable to handle compressible flow, the specie concentration, Y_i is only dependent on mass, and has the units kg/kg. The relation between the C_i and Y_i is;

$$Y_i = \frac{C_i}{\rho_k} \tag{63}$$

where ρ_k is the phase density.

The specie equation in multiphaseEulerFoam is solved inside the pimple loop, listing 4.4 line 111 (#include "YEqns .H"). The specie equation that are imported look like listing 4.5, the specie equation are solved for all species in all phases. The Yi.Eqn in listing 4.5 line 22 are defined in the MultiComponentPhaseModel.C [116], as show in listing 4.6. Due to the rewriting of specie concentration, the source terms are also rewritten. The reaction term, equation 50, is written as:

$$\dot{R}_{i,k} = k_i[A_g](1 - \theta(t))$$

where $[A_g]$ is the concentration of the gas A. Considering the rewriting in equation 63, the reaction source term is rewritten into equation 64, where ρ_k is the phase density.

$$\dot{R}_{i,k} = \rho_k k_i Y_A(1 - \theta(t)) \tag{64}$$

```
78  {
79      autoPtr<phaseSystem::specieTransferTable>
80          specieTransferPtr(fluid.specieTransfer());
81
82      phaseSystem::specieTransferTable&
83          specieTransfer(specieTransferPtr());
84
85      fluid.correctReactions();
86
87      forAll(fluid.multiComponentPhases(), multiComponentPhasei)
88      {
89          phaseModel& phase = fluid.multiComponentPhases()[multiComponentPhasei];
90
91          UPtrList<volScalarField>& Y = phase.YActiveRef();
92          const volScalarField& alpha = phase;
93          const volScalarField& rho = phase.rho();
```

```
94
95          forAll(Y, i)
96          {
97              fvScalarMatrix YiEqn
98              (
99                  phase.YiEqn(Y[i])
100                 ==
101                   *specieTransfer[Y[i].name()]
102                 + fvOptions(alpha, rho, Y[i])
103             );
104
105             YiEqn.relax();
106             YiEqn.solve("Yi");
107         }
108     }
109
110     fluid.correctSpecies();
111 }
```

Listing 4.5: The YiEqns in multiphaseEulerFoam. The solver calls upon the YiEqn in MulticomponentPhaseModel.C

```
144 {
145     const volScalarField& alpha = *this;
146     const surfaceScalarField alphaRhoPhi(this->alphaRhoPhi());
147     const volScalarField& rho = this->thermo().rho();
148
149     return
150     (
151         fvm::ddt(alpha, rho, Yi)
152       + fvm::div(alphaRhoPhi, Yi, "div(" + alphaRhoPhi.name() + ",Yi)")
153       + this->divj(Yi)
154      ==
155         alpha*this->R(Yi)
156
157       + fvc::ddt(residualAlpha_*rho, Yi)
158       - fvm::ddt(residualAlpha_*rho, Yi)
159     );
160 }
```

Listing 4.6: The definition of YiEqn in MulticomponentPhaseModle.C

The combined listings 4.5 and 4.6 makes out the specie equation 46 (this is a generic code that can handle turbulence), where specieTransfer[Y[i].name()] is the specie transfer term, $\dot{M}_{i,k}$ and alpha*this->R(Yi) is the reaction term, $\dot{R}_{i,k}$.

5 Model Development

There exist solvers that are capable of describing adsorption in a fluidized bed in the Eulerian-Lagrangian frame of reference. This makes them computational demanding, especially when it comes to large simulations. The foundation already exists in the OpenFOAM library for creating an adsorption model, as well as simulations with similar mechanisms to adsorption. MultiphaseEulerFOAM (In OpenFOAM v8; openfoam8/applications/solvers/multiphase/multiphaseEulerFoam [117]) was chosen because it uses the Eulerian-Eulerian frame of reference, it allows for chemical reactions and multi-specie.

The tutorial cases, found in
(/openfoam8/tutorials/multiphase/multiphaseEulerFoam/laminar/), that was used as the foundation were;

- Fluidized bed.
- BubbleColumnEvaporatingDissolving.
- TitaniaSynthesisSurface.

The basic idea behind this model is to use the chemical reaction to handle the reaction rate of the phase transfer. Then afterward, transfer the species between phases with the same rate as the chemical reaction. In the early stage of development, the system will be considered isotherm, so most of the thermal effects could be neglected and temperature-dependent properties set constant.

5.1 Model Description

The model will describe a system consisting of two phases, multiple species in both phases, one reaction resulting in a phase transformation, for a compressible system. The specification for the system must be inputted into OpenFOAM so the solver can access necessary input data.

The model considers two phases, a gas phase and a solid particle phase. These are defined by the alpha.gas and alpha.particles folders in the 0-directory figure 5.1. The initial flow of both phases in the system and the inlet velocity are defined in the U-files and temperature in the T-files. p and p_rgh define the pressure and hydrostatic pressure. There are only defined three initial species in the system (Al2O3, Ar, HF), the generated species, due to reaction, will have boundary conditions of Ydefault files according to the phase they are in.

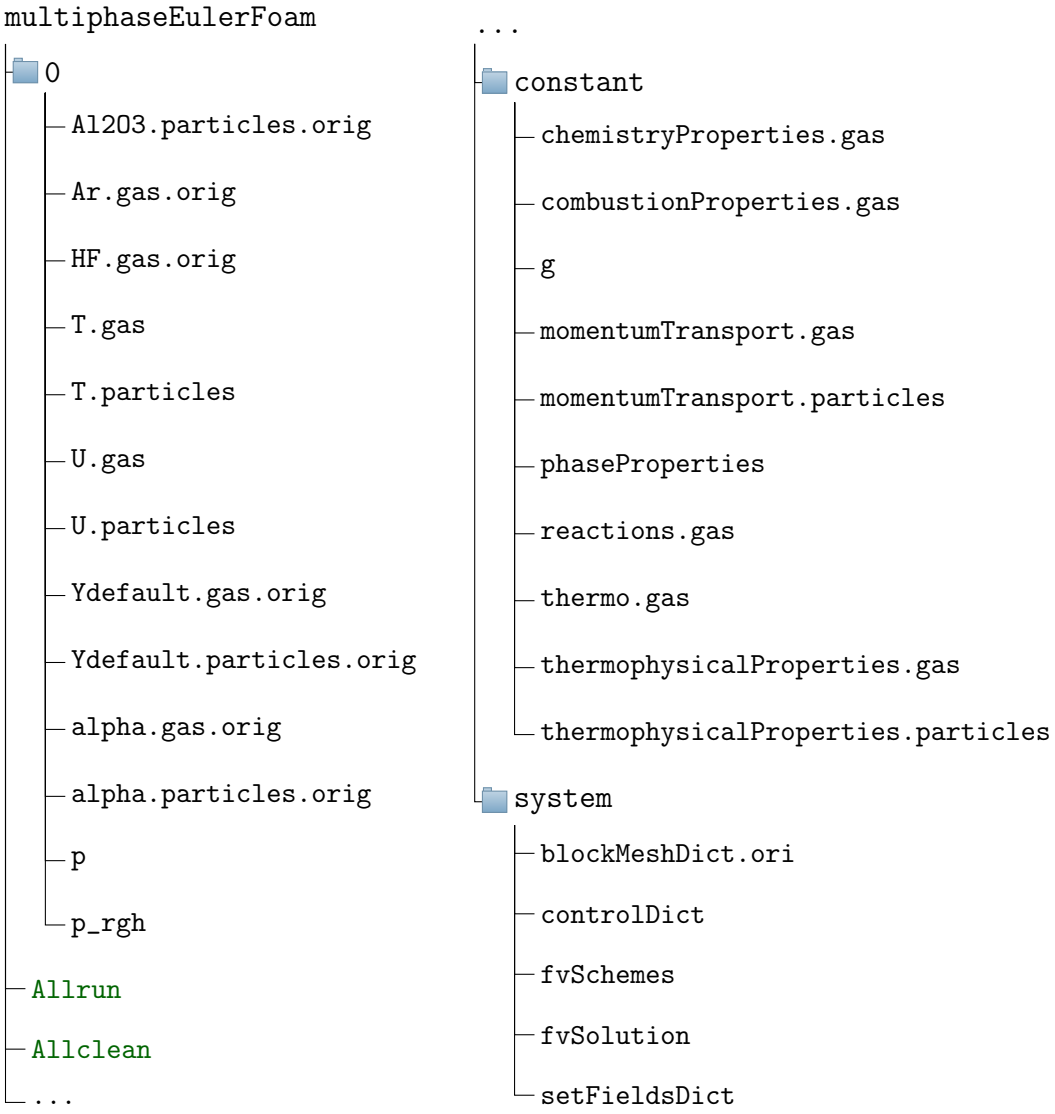


Figure 5.1: The files and the file structure used in the model

The properties of the different phases and species are defined in the constant folder. MomentumTransport dictates the flow regime, i.e laminar or turbulent (Reynolds Averaged Simulations (RAS) or Large Eddy Simulations (LES), while chemistry and combustion properties dictate the reaction system. Reactions.gas defines the reaction and reaction type, e.g. reversible/irreversible, as well as the reaction rate coefficient. thermophysicalproperties-files hold the thermophysical properties and the different models used to calculate them. The phaseProperties dictate how the different phases interact with each other and are possibly the most important file for a multiphase model. The system directory contains information on the system itself; blockMeshDict on the geometry and mesh, controlDict on simulation controls, like time step and end time, fvSchemes on the numerical schemes, such as derivatives, fvSolution contains equation solvers, tolerance and algorithms and the setFieldsDict overwrites the initial condition, allowing for better control when specifying the phase distribution.

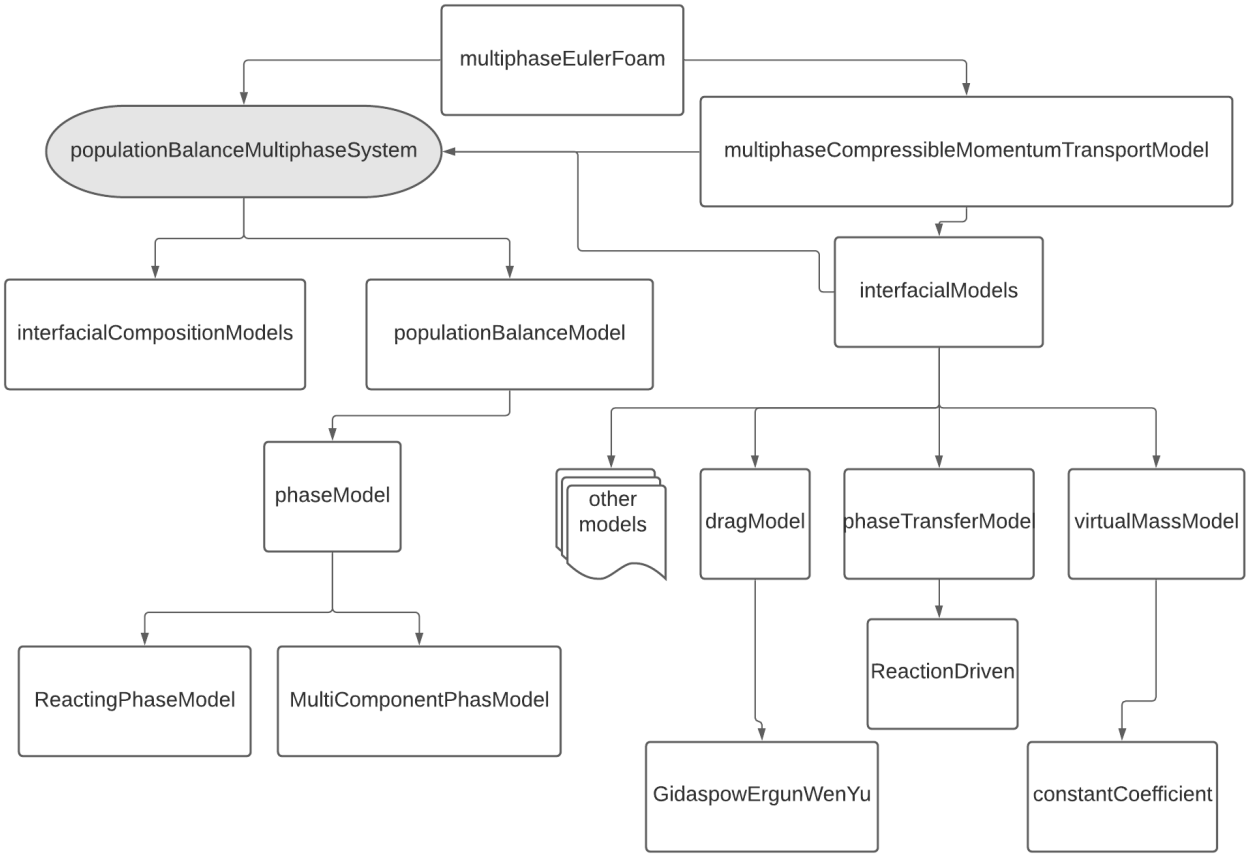


Figure 5.2: Shows some of the important files and classes called upon by multiphaseEulerFoam

Some of the important classes and systems called up by the solver are shown in figure 5.2. There are a lot more classes called upon, but this might make it easier to understand how to model is solved and structured.

multiphaseEulerFoam calls on the multiphaseCompressibleMomentumTransportmodel, this class's main task are to calculate the solid phase pressure. Unfortunately, all the models calculating the solid phase pressure are based on RAS, meaning that simulations using laminar or LES turbulence can't consider the particle-particle interaction. The interfacialModel class looks through the phase-Properties file, finding the interfacial model classes and the model calculating and return the source term to multiphaseEulerFoam. The populationBalanceMultiphaseSystem allows for multiple particle sizes using the populationBalanceModel and dividing the size groups into subdivisions and solving the population balance equation for each of the size fractions.

5.1.1 Reaction rate constant

The chemical reaction rate is calculated by Reaction.C (openfoam8/src/thermophysicalModels/specie/reaction/Reactions/Reaction/ [118]). The nature of the reversibility of the reaction is defined in the simulations reaction folder. If the type is irreversible, it should stand before the reaction rate constant type name, see listing 5.3.

The reaction, or more accurate, the reaction rate constant used the surfaceArrheniusReactionRate (/openfoam8/src/thermophysicalModels/chemistryModel/
reactions/surfaceArrheniusReactionRate [119]) was used as an basis, code in listing 5.1.

This modified reaction rate considers the surface area of the particle by calculating the surface area as;

$$k = (Aexp(-\frac{T_a}{T})a$$

The equation is the same as equation 53, but in OpenFOAM the activation energy is written as $T_a = \frac{E_a}{R}$. In addition, multiphaseEulerFoam calculates the reaction rate for a control volume, the surface area is therefore expressed as surface area pr volume. It is calculated by multiplying the volume fraction of the solid phase in the cell by the surface area pr volume, saved by storeA in the phaseproperties.H file.

In OpenFOAM, the code calculating the reaction rate constant is;

```
46 inline Foam::scalar Foam::surfaceArrheniusReactionRate::operator()
47 (
48     const scalar p,
49     const scalar T,
50     const scalarField& c,
51     const label li
52 ) const
53 {
54     return ArrheniusReactionRate::operator()(p, T, c, li)*aField_[li];
55 }
```

Listing 5.1: Defining the surfaceArrheniusReactionRate operator in adsorptionArrheniusReactionRateI.H

The modified reaction rate constant calculations take into consideration the concentration of available adsorption sites, (1-θ) in equation 55. Taking into consideration that the species in a phase is given as a weight fraction of the phase $(1-\theta)$ is calculated. In the code, listing 5.2, $(1-\theta)$ is calculated first as a scalar M and multiplied with the surface area dependent reaction rate coefficient;

```
63 inline Foam::scalar Foam::adsorptionArrheniusReactionRate::operator()
64 (
65     const scalar p,
66     const scalar T,
67     const scalarField& c,
68     const label li
69 ) const
70 {
71
72   scalar M = 1 - (((adsoField_[li]*pow(10,6))/(adsorbentField_[li]+VSMALL))/adsMax_);
73
74     return ArrheniusReactionRate::operator()(p, T, c, li)*aField_[li]*(M);
75 }
```

Listing 5.2: Modified surfaceArrheniusReactionRate, concidering the avaible adsorption cites to calculate the the reaction rate constant for the adsorption

where adsoField_[li] and adsorbentField_[li] are the weight fraction of the phase making up the adsorbate and adsorbent in the cell. adsMax_ is the adsorption capacity in mg adsorbate pr kg adsorbent. The +VSMALL at line 72 is there to ensure stability when calculating the reaction rate by adding a small number (1e-300) to the adsorbentField value, so it never is zero.

The code is modified to take in the necessary information, through the reaction header-file in the case setup. The reaction header-file should look something like listing 5.3 when using the modified reaction rate constant code.

```
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

reactions
{
    Name    //-Name of reaction, example; AdsorptionReaction
    {
        type        irreversibleAdsorptionArrhenius; //-name of this reaction

        reaction    "gas = gas_s";  //-Reaction of gas to the adsorbed gas

        A           __;     //- Pre-exponential factor
        beta        __;     //- if the reaction is
        Ta          __;     //- Activation energy / universal gas constant
        a           __;     //- naming the surface area pr volume, a + .phase
        adsMax      __;     //- capacity in mg adsorbate/kg adsorbent
        adso        __;     //- Name of adsorbate field + .phase
        adsorbent   __;     //- Name of adsorbent field + .phase
    }
}

// ************************************************************************* //
```

Listing 5.3: Shows how to use irreversibleAdsorptionArrhenius

The complete code for AdsorptionArrhenius is in appendix C, as well as on GitHub: Link to GitHub.

5.1.2 Phase transfer

The phase transfer, $I_{I,k}$ in equation 46 is calculated by the phaseTransferModel.C (/openfoam8/applications/solvers/multiphase/multiphaseEulerFoam/interfacialModels/phaseTransferModels/phaseTransferModel [120]). This code write the total mass transfer between phases as (in the code; dmdtf) and specie mass transfer transfer (in the code;dmidtf). dmdtf and dmidtf is calculated in the modified reactionDriven phase transfer listing 5.4.

```
Foam::HashPtrTable<Foam::volScalarField>
Foam::phaseTransferModels::reactionDriven::dmidtf() const
{
    HashPtrTable<volScalarField> result;

    forAll(species_, i)
    {
        const word name = species_[i];

```

```
93          volScalarField& Y =
94              const_cast<volScalarField&>(reactingPhase_.Y(name));
95
96          result.set
97          (
98              species_[i],
99              (sign_*reactingPhase_*reactingPhase_.R(Y) & Y).ptr()
100         );
101     }
102
103     return result;
104 };
```

Listing 5.4: phase transfer rate calculated in reactionDriven

The modification is done to line 99 in listing 5.4, the line is changed to the line shown in lisiting 5.5

```
99              (sign_*reactingPhase_.R(Y) & Y).ptr()
```

Listing 5.5: Shows the modified code line in myReactionDriven.

This small change in the code makes the phase transfer rate a function of only the Reaction rate, without multiplying it with the phase fraction, as this has already been done when calculating the reaction rate.

The code works by calculating the specie transfer rate for all species given in the input in the solver. It calculates the specie transfer rate as shown in listing 5.5, where the sign_ is 1 or -1, depending on the direction of the transfer, this is multiplied by the reaction rate source term reactingPhase_.R(Y), and stored so it is available for the populationBalanceMultiphase class or any of the other classes used defined the phaseProperties file.

The complete code for the modified reactionDriven is in appendix D, as well as on GitHub: Link to GiHub.

6 Simulations

All the simulations are done in two dimensions to reduce the computational time. The same schemes are used for all simulations, table 6.1 gives an overview of these. The Solvers tolerances, PIMPLE correctors and relaxation factors are for the most cases kept unchanged. In some simulations the the tolerance or number of correctors might be changed to achieve a stable simulation, this will be specified in the case setup. The standard fvSolution setup is shown in table 6.2. for simulation done in chapter 6.1, 6.2 the controlDict file is the same, the most important parameters from the controlDict-files are listed in table 6.3.

Table 6.1: Shows the schemes used in the simulations.

ddtSchemes	
default	Euler;
gradSchemes	
default	Gauss linear;
divSchemes	
default	none;
"div\(phi,alpha.*\)"	Gauss vanLeer
"div\(phir,alpha.*\)"	Gauss vanLeer
"div\(alphaRhoPhi.*,U.*\)"	Gauss limitedLinearV 1;
"div\(phi.*,U.*\)"	Gauss limitedLinearV 1;
"div\(alphaRhoPhi.*,(h|e).*\)"	Gauss limitedLinear 1;
"div\(alphaRhoPhi.*,K.*\)"	Gauss limitedLinear 1; (not neccesary)
"div\(alphaRhoPhi.*,Y.*\)"	Gauss limitedLinear 1;
"div\(alphaPhi.*,p\)"	Gauss limitedLinear 1;
"div\(\(\(\(alpha.**thermo:rho.*\)* nuEff.*\)*dev2\(T\(grad\(U.*\)\)\)\)\)"	Gauss linear;
laplacianSchemes	
default	Gauss linear uncorrected;
bounded	Gauss linear uncorrected;
interpolationSchemes	
default	linear;
fi	upwind alphaRhoPhi.particles; (for multiple d_p)
snGradSchemes	
default	corrected;
bounded	uncorrected;

Table 6.2: Shows the solution algorithms used to in the simulations.

Solvers		PIMPLE	
"alpha.*"		nOuterCorrectors	3;
nAlphaCorr	1;	nCorrectors	2;
nAlphaSubCycles	3;	nEnergyCorrectors	1;
implicitPhasePressure	yes;	nNonOrthogonalCorrectors	0;
extremaCoeff	1;	faceMomentum	yes;
solver	PBiCGStab;	RelaxationFactors	
preconditioner	DIC;	**equations**	
tolerance	1e-18;	".*"	1.0;
relTol	0;		
minIter	1;		
p_rgh			
solver	GAMG;		
smoother	DIC;		
tolerance	1e-10;		
relTol	0.01;		
p_rghFinal			
$p_rgh;			
relTol	0;		
"U.*"			
solver	smoothSolver;		
smoother	symGaussSeidel;		
tolerance	1e-7;		
relTol	0;		
minIter	1;		
"(h|e).*"			
solver	smoothSolver;		
smoother	symGaussSeidel;		
tolerance	1e-8;		
relTol	0;		
minIter	1;		
maxIter	10;		
"(Yi|f|kappa).*"			
solver	PBiCGStab;		
preconditioner	DILU;		
tolerance	1e-20;		
relTol	0;		
residualAlpha	1e-12;		
minIter	1;		

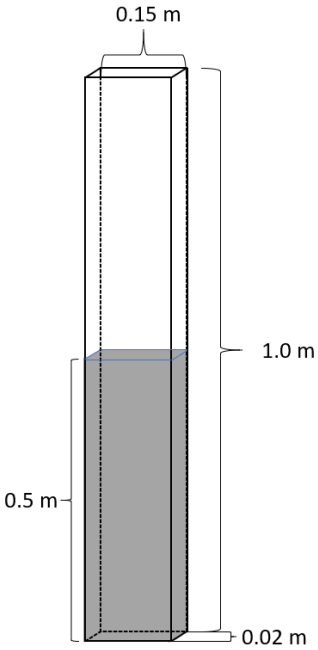
Table 6.3: Lists the most important parameters from the controlDict-file used in most of the simulations

Parameter	Input
application	multiphaseEulerFoam;
stopAt	endTime;
endTime	8;
writeInterval	0.01;
deltaT	2e-4;
runTimeModifiable	off;
adjustTimeStep	yes;
maxCo	0.7;
maxDeltaT	0.5;

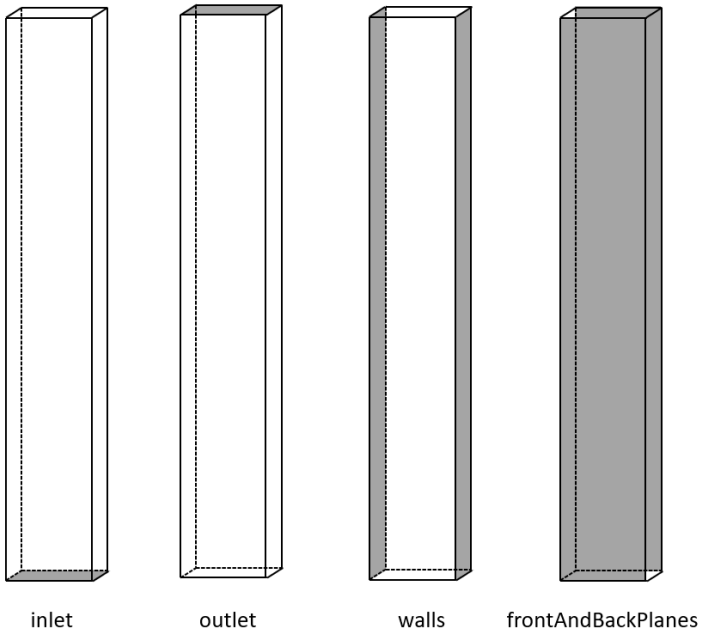
6.1 Case 1: No phase transfer or reaction

The first case is to see if the multiphase simulation works without the reaction (and phase transfer), to see how the particles and bed behaves.

The system is 1 meter high, and 0.15m wide. The bottom of the system is filled with particles up to a height 0f 0.5 m, sketch in figure 6.1a. The different patches and their name is shown in figure 6.1b, and are used when defining the boundary conditions. The different material parameters is listed in table 6.4, and the initial and boundary conditions are listed in table 6.5.



(a) Shows the dimensions for the case setup and the grey area represents the particles in the system.



(b) Shows the patch names of the system and where they apply

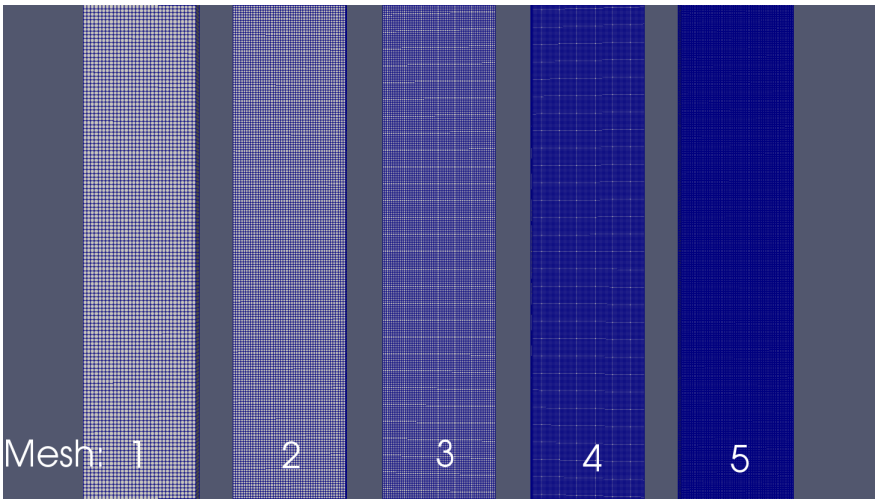
Table 6.4: Shows the physical and thermophysical properties of the system simulated in case 1.

Simulation parameters	Value	Units	Source
	Alumina.particles		
Density	2670	kg/m^3	[3]
Molar mass	101.96	g/mol	
Particle diameter	$125 \cdot 10^{-6}$	m	
	Bed properties		
Maximum solid packing	0.55		
Initial solid packing	0.55		
Bed height, settled	0.5	m	
	HF.gas		
Density	Perfect gas		
Molar mass	20.01	g/mol	
T_{low}	1200	K	HSC [121]
T_{high}	3300	K	HSC [121]
T_{common}	1200	K	HSC [121]
highCpCoeffs	(29.947, 5.829, -2.861, -0.693, 0)		HSC [121]
lowCpCoeffs	(29.947 , -3.137, -0.152, 3.365, 0)		HSC [121]
A_s	$1.69 \cdot 10^{-6}$		Calculated from [122]
T_s	402		Calculated from [122]
	Ar.gas		
Density	Perfect gas		
Molar mass	39.948	g/mol	
T_{low}	100	K	HSC [121]
T_{high}	11000	K	HSC [121]
T_{common}	8000	K	HSC [121]
highCpCoeffs	(33.455, -1.986, -1544,132, 0.088, 0)		HSC [121]
lowCpCoeffs	(20.786, 0, 0, 0, 0)		HSC [121]
A_s	$2.038169524 \cdot 10^{-6}$		Calculated from [123]
T_s	165		Calculated from [123]

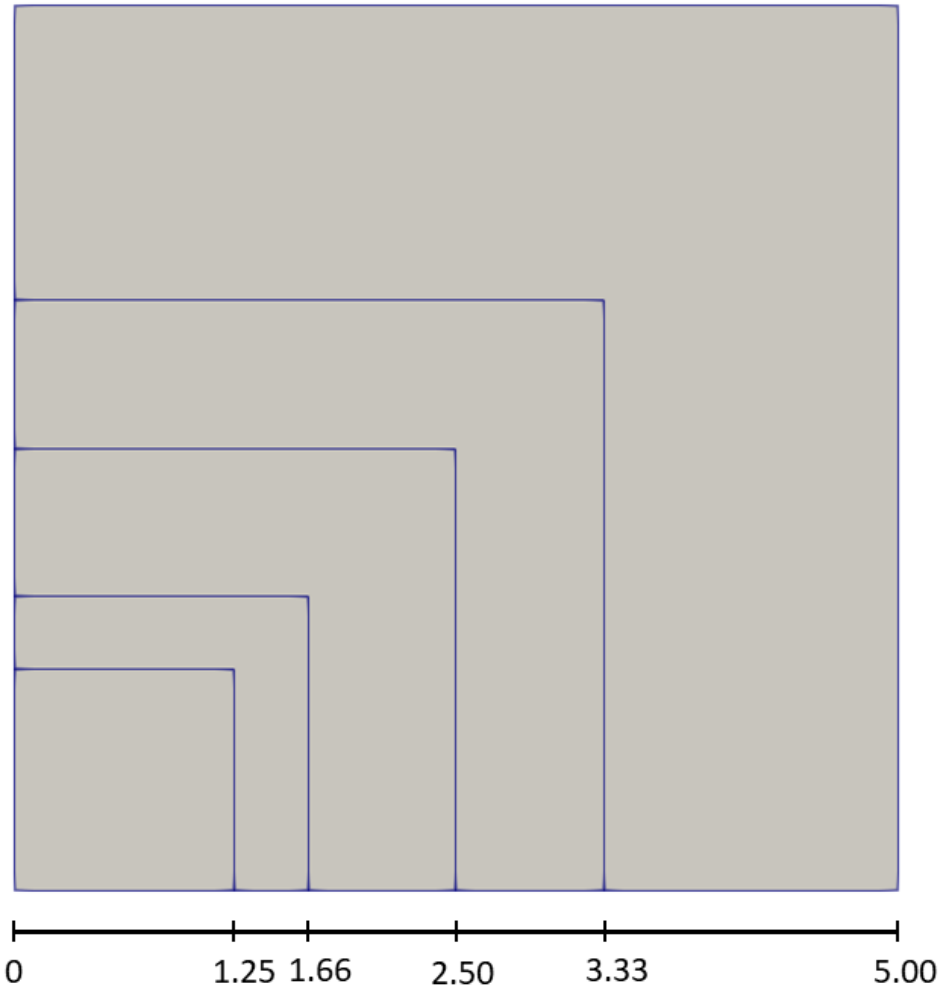
The system simulates a gas flow from the bottom, with a superficial velocity of 0.1 m/s and a 90/10 volume fraction Argon/HF in the inlet and the system is isotherm at 400 K. The four different meshes can be seen in figure 6.2, while information about the mesh can be found in table 6.6. For all simulation the Courant number was set to not exceed 0.7

Table 6.5: Shows the boundary and initial conditions for Cases 1. The .p and .g endings are Abbreviations for .particles and .gas respectively. The alpha fields are the phase fields, T are the temperature fields, U are the velocity fields, p is the pressure field, p_rgh is the hydrostatic pressure field and Al2O3, Ar and HF are specie fields.

	Al2O3.p	**Ar.g**	**HF.g**
InternalField	uniform 1	uniform 1	uniform 0
Inlet	fixedValue; uniform 0.0;	fixedValue; uniform 0.99;	fixedValue; uniform 0.1;
outlet	zeroGradient;	zeroGradient;	zeroGradient;
walls	zeroGradient;	zeroGradient;	zeroGradient;
	T.g	**T.p**	**U.g**
InternalField	uniform 400;	uniform 400;	uniform (0 0 0)
Inlet	fixedValue; uniform 400;	zeroGradient;	interstitialInletVelocity; uniform (0 0.1 0); alpha.gas; $internalField;
outlet	inletOutlet; phi.gas; uniform 400;	inletOutlet; phi.gas; uniform 400;	pressureInletOutletVelocity; phi.gas; $internalField;
walls	uniform 400; zeroGradient;	uniform 400; zeroGradient;	noSlip
	U.p	**Ydefaul.g**	**Ydefaul.p**
InternalField	uniform (0 0 0)	uniform 0.0;	uniform 0.0;
Inlet	fixedValue; uniform (0 0 0);	zeroGradient	fixedValue; uniform (0 0 0);
outlet	fixedValue; uniform (0 0 0);	zeroGradient	fixedValue; uniform (0 0 0);
walls	noSlip	zeroGradient	zeroGradient;
	alpha.g	**alpha.p**	**p**
InternalField	uniform 1;	uniform 1;	uniform 1e5;
Inlet	zeroGradient	zeroGradient	fixedValue; uniform 1e5
outlet	zeroGradient	zeroGradient	calculated; $internalField;
walls	zeroGradient	zeroGradient	calculated; $internalField;
	p_rgh		
InternalField	uniform 1e5;		
Inlet	fixedFluxPressure; $internalField;		
outlet	prghPressure; $internalField;		
walls	$internalField; fixedFluxPressure; $internalField;		



(a) mesh



(b) The different cells, bottom scale is in mm

Figure 6.2: Shows the difference in refined mesh and the cell size

Table 6.6: Shows the mesh name, length of on cell in the system and the total amount of cells

Mesh	nCells direction	Cell side length [cm]	Number of cells
1	(30 200 1)	5.00	6000
2	(45 300 1)	3.33	13500
3	(60 400 1)	2.50	24000
4	(90 600 1)	1.66	54000
5	(120 800 1)	1.25	96000

In addition to the simulations investigating the mesh dependence additional simulation was conducted to investigate the volume of the particle phase

To investigate the increase in volume, additional simulations was completed to investigate the volume changes in the particles phase. Simulation C1.1-C1.5 in table 6.7 is the simulation with different mesh. Table 6.7 gives an overview of the different simulations that was completed and the changes done to the models.

Table 6.7: Gives and overview of the simulation conducted and the changes done to the simulations.

Case 1	Mesh	Changes	Co	PIMPLE(Outer,inner)	nAlphacorr(outer,inner)
C1.0	1	no specie	0.9	3,2	1,3
C1.1	1	-	0.7	3,2	1,3
C1.2	2	-	0.7	3,2	1,3
C1.3	3	-	0.7	3,2	1,3
C1.4	4	-	0.7	3,2	1,3
C1.5	5	-	0.7	3,2	1,3
C1.6	1	-	0.7	2,2	2,3
C1.7	1	-	0.1	3,2	1,3
C1.8	1	BC	0.7	3,2	1,3
C1.9	1	BC	0.7	3,2	1,3
C1.10	1	BC	0.7	3,2	1,3
C1.11	1	-	0.7	5,5	5,5

In simulation C1.0 the species was removed. The particle phase (alpha.particles) has the same thermophysical properties as the other simulations with Al2O3 inside the phase, while the gas phase (alpha.gas) is slightly altered as the gas phase has the properties as pure argon gas. Simulation C1.1-C1.5 is described above. Simulation C1.6 has the same mesh size as simulation C1.1, but the number of outer PIMPLE loops are reduced from 3 to two. Simulation C1.7 has reduced the maximum Courant number, decreasing the time steps. In simulation C1.8- C1.10 some of the boundary conditions are changed. The overview of the changed boundary conditions are given in table 6.8. Simulation C1.11 has increased amount of outer and inner PIMPLE and MULES loops.

Table 6.8: Shows the changed boundary condition, in what file the change is made and for which simulation the change is done .

Simulations	File	Change

C1.8	Al2O3	inlet
		{
		type inletOutlet;
		inletValue uniform 0.0;
		value uniform 0.0;
		}

C1.9	U.gas	inlet
		{
		type interstitialInletVelocity;
		inletVelocity uniform (0 0 0);
		alpha alpha.gas;
		value $internalField;
		}

C1.10	Al2O3	inlet
		{
		type zeroGradient;
		}

	alpha.p	inlet
		{
		type zeroGradient;
		}
		outlet
		{
		type zeroGradient;
		}

6.2 Case 2: Phase transfer, no adsorption limit

Simulation with a fluidized bed with phase transfer. The initial parameters and boundary conditions are the same as in Case 1 (chapter 6.1), table 6.5. The simulation is done for 2 different densities for the solid phase, the thermophysical properties are found in table 6.9. The adsorption capacity is set to 10^30, making it in practice disabled. The meshes used is mesh 1 and 3 in table 6.6, with a cell side length of 5.00 and 2.50 mm. The reaction parameters are listed in table 6.10.

Table 6.9: Overview of the thermophysical properties for case 2. Density 1 and 2 are for two different simulations.

Simulation parameters	Value	Units	Source
	Alumina.particles		
Density 1	2670	kg/m^3	[3]
Density 2	2000	kg/m^3	
Molar mass	101.96	g/mol	
Average particle diameter	$125 \cdot 10^{-6}$	m	
	HF_s.particles		
Density 1	2670	kg/m^3	
Density 2	2000	kg/m^3	
Molar mass	20.01	g/mol	
	Bed properties		
Maximum solid packing	0.55		
Initial packing	0.55		
Bed height, settled	0.5	m	
	HF.gas & HF_s.gas		
Density	Perfect gas		
Molar mass	20.01	g/mol	
T_{low}	1200	K	HSC [121]
T_{high}	3300	K	HSC [121]
T_{common}	1200	K	HSC [121]
highCpCoeffs	(29.947, 5.829, -2.861, -0.693, 0)		HSC [121]
lowCpCoeffs	(29.947 , -3.137, -0.152, 3.365, 0)		HSC [121]
A_s	$1.69 \cdot 10^{-6}$		Calculated from [122]
T_s	402		Calculated from [122]
	Ar.gas		
Density	Perfect gas		
Molar mass	39.948	g/mol	
T_{low}	100	K	HSC [121]
T_{high}	11000	K	HSC [121]
T_{common}	8000	K	HSC [121]
highCpCoeffs	(33.455, -1.986, -1544,132, 0.088, 0)		HSC [121]
lowCpCoeffs	(20.786, 0, 0, 0, 0)		HSC [121]
A_s	$2.038169524 \cdot 10^{-6}$		Calculated from [123]
T_s	165		Calculated from [123]

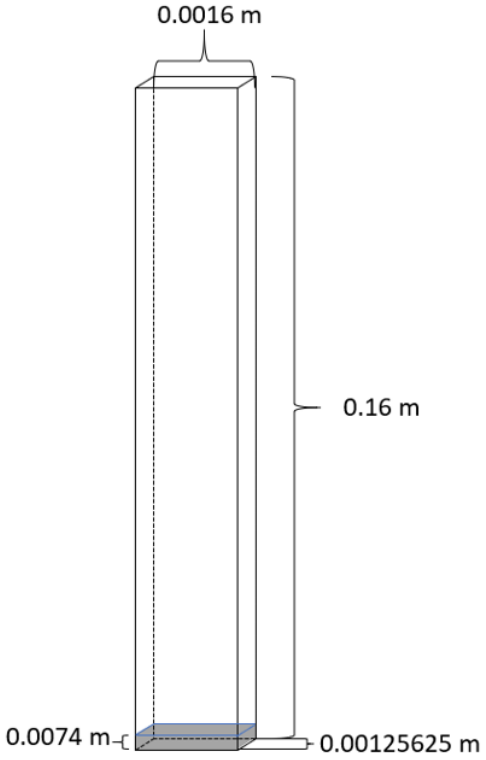
Table 6.10: Reaction inputs, into the custom reaction rate coefficient calculator, irreversibleAdsorptionArrhenius, for case 2

Parameter	Input
type	irreversibleAdsorptionArrhenius;
reaction	"HF = HF_s";
A	8e10;
beta	0.0;
Ta	1e4;
a	a.particles;
adsMax	1e30;
adso	HF_s.particles;
adsorbent	Al2O3.particles;

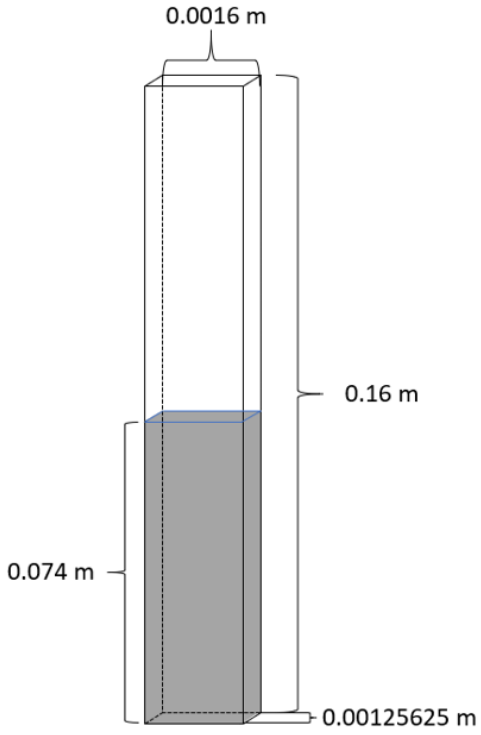
6.3 Case 3: Phase transfer

Simulation inspired by Agbenyegah's [3] experiments. The temperature is fixed at 353 K and an HF gas-concentration of 650 mg/Nm^3 to show how the simulation can be set up.

The reactor in the experiment consisted of a tubular reactor, with an internal diameter of 16 mm and a height of 160 mm. The 2D sketch of the translated reactor in OpenFOAM is shown in figure 6.3. The cross-sectional area is the same in the simulation as in the experiment. In addition to running the simulation with a bed height of 0.0074 m, two simulations with a bed height at 0.074 m and temperatures of 353 and 373 K are run to investigate the adsorption at different temperatures. The simulation at 373 K has a different adsorption capacity and inlet velocity and is shown in table 6.14. The mesh is consists of 9000 cells (30 300 1), making the cell length 0.53 mm.



(a) Dimensions for simulations with low bed height



(b) Dimensions for simulations with high bed height

Figure 6.3: Schematic of experiment conducted by Agbenyegah [3], transformed into a system that can be simulated in 2D with different bed heights.

The reaction rate for the adsorption is calculated from Agbenyegah's [3] thermometric titration, calculations in appendix A.1, and the calculation for the inlet flow is in appendix A.2. The adsorption capacity is calculated by using Agbenyegah's regression model in equation 2, with the water content at zero and temperature at 353 K and 373K. In addition, the adsorption capacity is divided by 10^4, to reduce the simulation time. Simulation from Case 2 has shown that the model generates about twice as much adsorbed HF as it should. To balance this the adsorption capacity is multiplied by 2.

The gas used in the experiment was N_2 gas, so Ar.gas is substituted with N2.gas. The boundary conditions for the simulations at 353 K are shown in table 6.13. The simulation at 373 K has similar boundary conditions, but there are done changes small changes. The gas inlet velocity is 0.0565 and the temperature is 373 K, the changed boundary conditions are shown in table 6.14. The reaction inputs are shown in table 6.11.

Table 6.11: Reaction inputs for the custom reaction rate coefficient calculator, irreversibleAdsorptionArrhenius, for simulations in Case 3

Parameter	Input
type	irreversibleAdsorptionArrhenius;
reaction	"HF = HF_s";
A	5.55e17;
beta	0.0;
Ta	18040;
a	a.particles;
adsMax (353 K)	4.3;
adsMax (373 K)	3.88;
adso	HF_s.particles;
adsorbent	Al2O3.particles;

The controlDict files are slightly modified from case 1 and 2. The file is shown in table 6.12. The thermophysical properties, and the bed properties are shown in table 6.15. The schemes and solution is the same as in case 1 and 2, table 6.1 and 6.2.

Table 6.12: Lists the most important parameters from the controlDict-file used in case 3

Parameter	Input
application	multiphaseEulerFoam;
writeInterval	0.1;
deltaT	2e-5;
runTimeModifiable	off;
adjustTimeStep	yes;
maxCo	0.7;
maxDeltaT	0.5;

Table 6.13: Shows the boundary and initial conditions for the simulations in Cases 3.

	Al2O3.p	**N2.g**	**HF.g**
InternalField	uniform 1	uniform 1	uniform 0
Inlet	fixedValue; uniform 0.0;	fixedValue; uniform 0.9995;	fixedValue; uniform 0.0005;
outlet	zeroGradient;	zeroGradient;	zeroGradient;
walls	zeroGradient;	zeroGradient;	zeroGradient;
	T.g	**T.p**	**U.g**
InternalField	uniform 353;	uniform 353;	uniform (0 0 0)
Inlet	fixedValue; uniform 353;	zeroGradient;	interstitialInletVelocity; uniform (0 0.0535 0); alpha.gas; $internalField;
outlet	inletOutlet; phi.gas; uniform 353;	inletOutlet; phi.gas; uniform 353;	pressureInletOutletVelocity; phi.gas; $internalField;
walls	uniform 353; zeroGradient;	uniform 353; zeroGradient;	noSlip
	U.p	**Ydefaul.g**	**Ydefaul.p**
InternalField	uniform (0 0 0)	uniform 0.0;	uniform 0.0;
Inlet	fixedValue; uniform (0 0 0);	zeroGradient	zeroGradient
outlet	fixedValue; uniform (0 0 0);	zeroGradient	zeroGradient
walls	noSlip	zeroGradient	zeroGradient;
	alpha.g	**alpha.p**	**p**
InternalField	uniform 1;	uniform 1;	uniform 1e5;
Inlet	zeroGradient	zeroGradient	fixedValue; uniform 1e5
outlet	zeroGradient	zeroGradient	calculated; $internalField;
walls	zeroGradient	zeroGradient	calculated; $internalField;
	p_rgh		
InternalField	uniform 1e5;		
Inlet	fixedFluxPressure; $internalField;		
outlet	prghPressure; $internalField; $internalField;		
walls	fixedFluxPressure; $internalField;		

Table 6.14: Shows the altered boundary and initial conditions for the simulation at 373 K. The other boundary conditions are the same as in table 6.13

	T.g	**T.p**	**U.g**
InternalField	uniform 373;	uniform 373;	uniform (0 0 0) interstitialInletVelocity;
Inlet	fixedValue; uniform 373;	zeroGradient;	uniform (0 0.0565 0); alpha.gas; $internalField;
outlet	inletOutlet; phi.gas; uniform 373;	inletOutlet; phi.gas; uniform 353;	pressureInletOutletVelocity; phi.gas; $internalField;
walls	uniform 373; zeroGradient;	uniform 373; zeroGradient;	noSlip

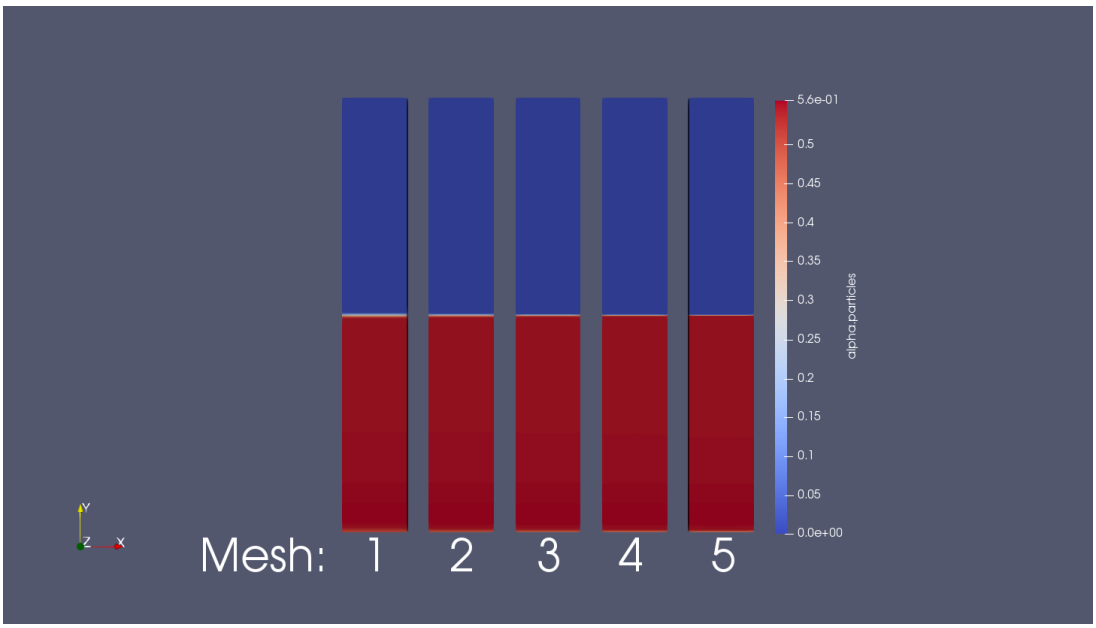
Table 6.15: Overview of the thermophysical properties for case 3

Simulation parameters	Value	Units	Source
	Alumina.particles		
Density 1	2670	kg/m^3	[3]
Molar mass	101.96	g/mol	
Average particle diameter	$70 \cdot 10^{-6}$	m	
	HF_s.particles		
Density 1	2670	kg/m^3	
Molar mass	20.01	g/mol	
	Bed properties		
Maximum solid packing	0.67		[3]
Initial packing	0.67		
Bed height, settled	0.0074	m	[3]
	HF.gas & HF_s.gas		
Density	Perfect gas		
Molar mass	20.01	g/mol	
T_{low}	1200	K	HSC [121]
T_{high}	3300	K	HSC [121]
T_{common}	1200	K	HSC [121]
highCpCoeffs	(29.947, 5.829, -2.861, -0.693, 0)		HSC [121]
lowCpCoeffs	(29.947 , -3.137, -0.152, 3.365, 0)		HSC [121]
A_s	$1.69 \cdot 10^{-6}$		Calculated from [122]
T_s	402		Calculated from [122]
	Ar.gas		
Density	Perfect gas		
Molar mass	28.01	g/mol	
T_{low}	100	K	HSC [121]
T_{high}	700	K	HSC [121]
T_{common}	350	K	HSC [121]
highCpCoeffs	(27.753, 0.605, 0.728, 4.960, 0)		HSC [121]
lowCpCoeffs	(29.298, -1.567, -0.007, 3.419, 0)		HSC [121]
A_s	$1.40 \cdot 10^{-6}$		Calculated from [123]
T_s	107		Calculated from [123]

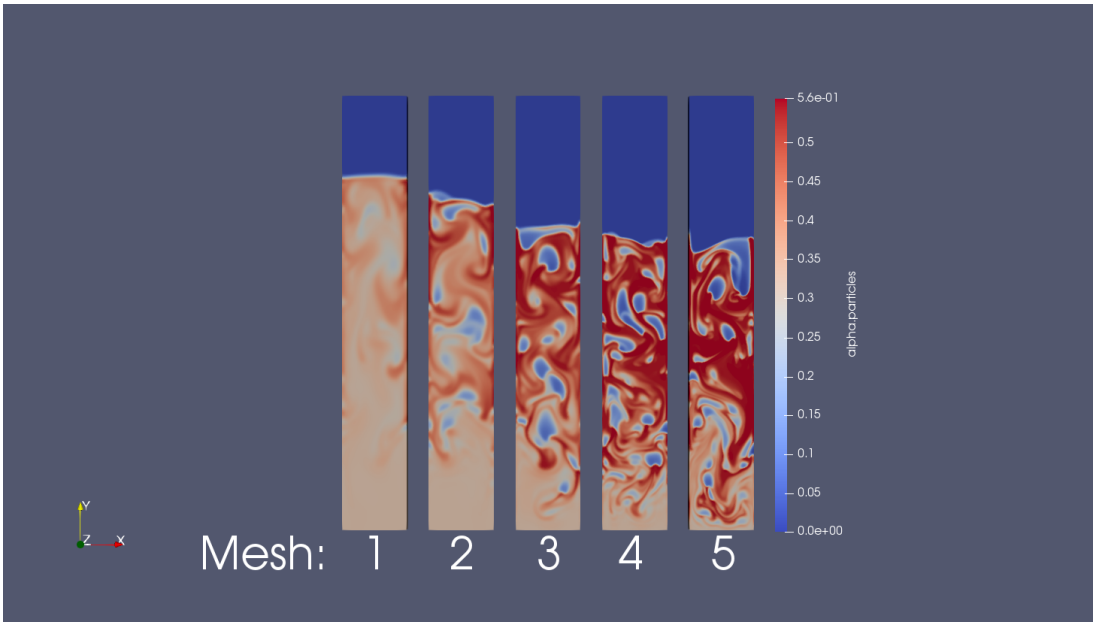
7 Results

7.1 Case 1: No phase transfer or reaction

To investigated how big impact the mesh have on the simulations, the system was simulated using 5 different meshes. Figure 7.1 shows the simulation at t=0 and after t=8 seconds. The simulation have no specie transfer or phase transfer, only momentum transfer between the phases.



(a) Shows the bed at t=0s



(b) Shows the bed at t=8s

Figure 7.1: Shows the difference in refined mesh and the cell size [video].

From figure 7.1 the bed height is closer to the initial bed height with a more refined mesh. With the coarsest mesh (mesh1), the particle phase appear to be more uniform then the other meshes, with almost none distinct formation of gas bubbles. Mesh 2 has more defined gas bubbles formed in the bed, and has a tendency to a higher volume fraction solid particles in the top part of the bed compared to the simulation with mesh 1. Mesh 3, 4 and 5 has similar bed height and comparable solid particles volume fraction distribution in the bed.

Case 1 consists of 5 main simulations with different mesh, as well as additional 6 simulations with the coarsest mesh, but with changes to e.g. Courant number, boundary conditions and species. overview in table 6.7.

The theoretical amount of alumina in the system is $(l \cdot b \cdot h \cdot \text{alpha} \cdot \text{particles} \cdot \rho_{particles})$ 2.2075 kg, and the theoretical volume is $8.25{\cdot}10^{-4}$. The amount of alumina in the system is plotted over time for the different meshes in figure 7.2. The particle volume, as a function of time, is plotted in figure 7.3. For all five meshes the mass of alumina and volume of the particle phase increases initially. Then they decrease, but are still above the theoretical values. After the initial increase and decrease, the system have an increase in mass and volume of the particle phase until the simulation ends at 8 seconds.

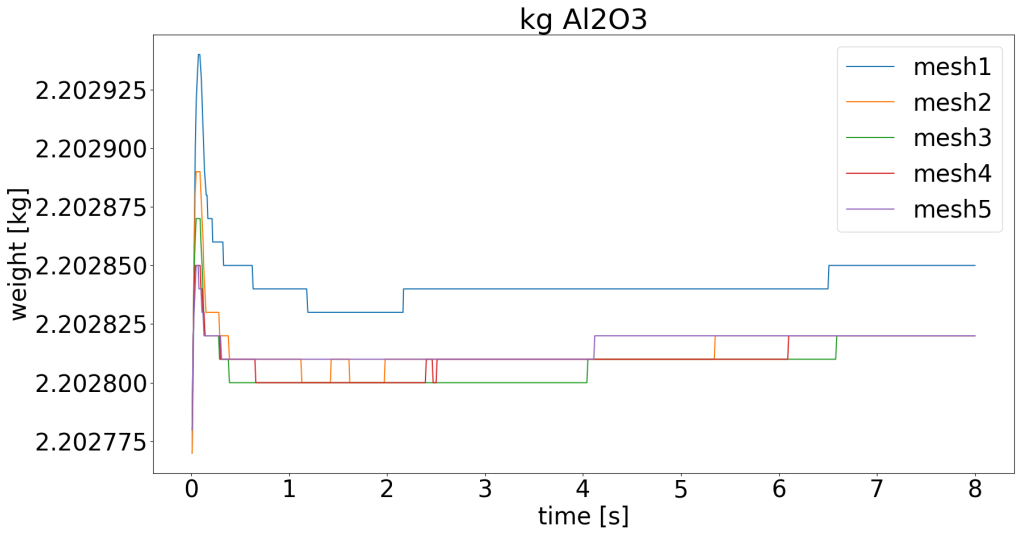


Figure 7.2: Shows the calculated mass of alumina in the system as a function of time for the 5 different mesh.

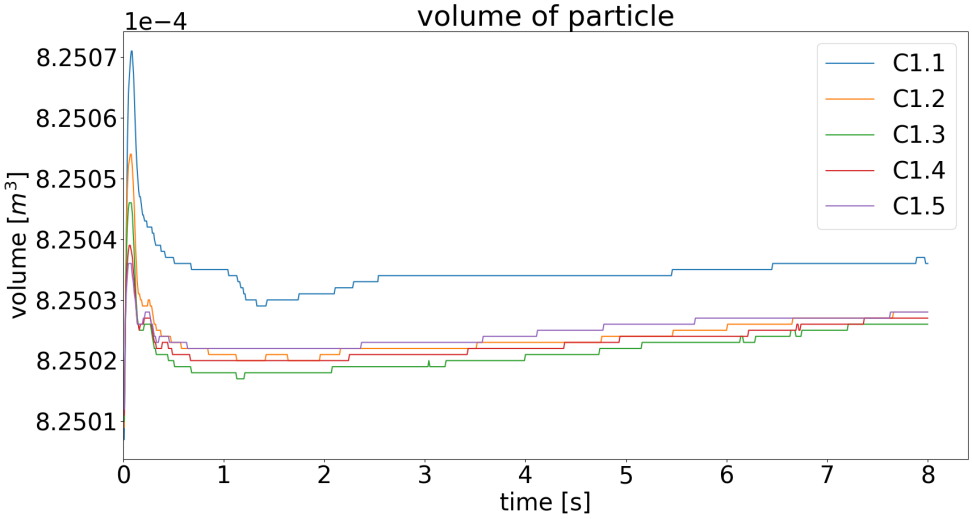


Figure 7.3: Shows the calculated volume of the solid phase in the system as a function of time for the 5 different mesh.

The change in the particle phase is shown in figure 7.4, for the 11 different simulation without reactions or phase transfer, listed in table 6.7. The simulation without species, C1.0, shows small variation in the volume fraction of the particle phase and is close to the theoretical volume. Simulation without flow in the system, C1.9, have an initial increase in the particle volume fraction, but after 0.5 seconds the volume appears constant in figure 7.4. Simulation C1.6, with 2 outer PIMPLE loops, reduced from 3, has the biggest deviation in the particle volume. Simulation C1.8, C1.7 and C1.11 is covered by C1.1, and there are no visual differences between them.

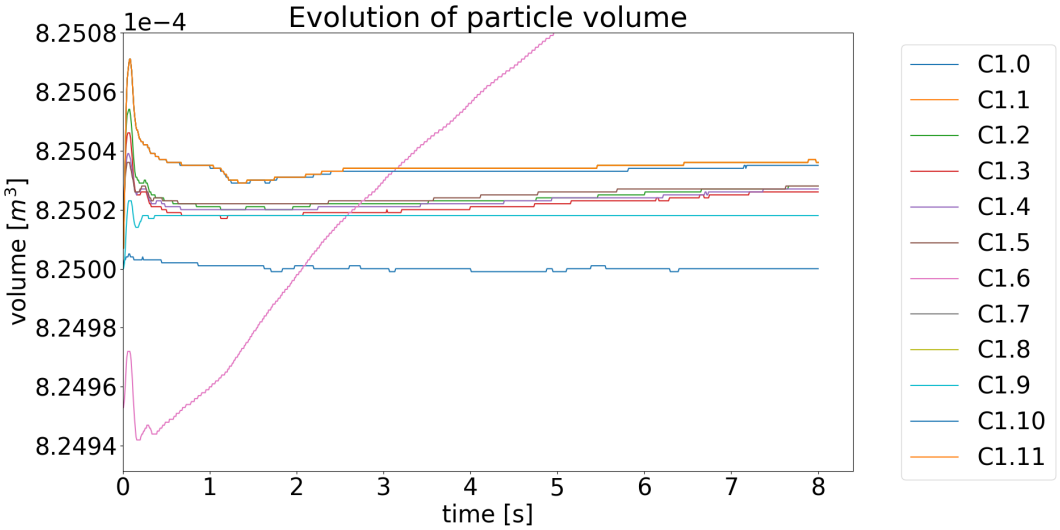


Figure 7.4: Shows the calculated volume of the particle phase as a function of time for the simulation in case 1.

7.2 Case 2: Phase transfer, no adsorption limit

Case 2 consists of four simulations. Two simulations with a course and fine mesh with a particle density of 2670 kg m^{-3}, and two simulations with the same course and fine mesh with a particle density of 2000 kg m^{-3}.

The mass balance and flow rate was calculated using the method in appendix B in Paraview. The inlet mass flow of HF.gas is shown in figure 7.5. The theoretical mass flow of HF.gas into the system was calculated to be $3.42{\cdot}10^{-5}$ kg s^{-1}. The mass flow of HF into the system are oscillating around $3.42{\cdot}10^{-7}$ kg s^{-1} for the fine mesh. For the simulation with lowest density the amplitudes are slightly larger, compared to the denser phase. The inlet flow for the courser mesh are not oscillating, but have a constant value of $3.42{\cdot}10^{-7}$.

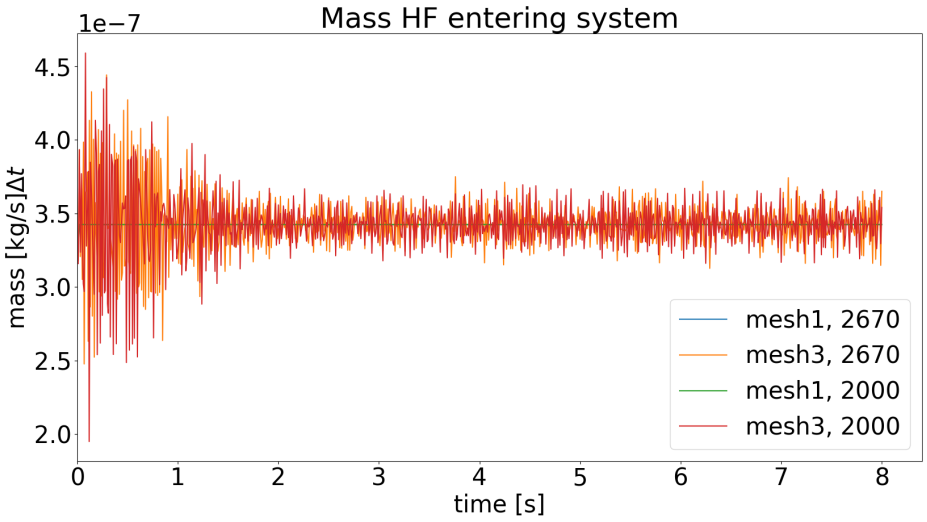


Figure 7.5: Shows the mass flow into the system for each time written.

Figure 7.6 shows the mass balance for the four simulations, with different density in the solid phase and different mesh. The mass balance is calculated for each time written, i.e the difference between the amount of HF in the system and the amount that flows out compared to what flows in to the system. The finest mesh with the lightest density has the biggest deviation in the calculated mass balance. With a courser mesh, the deviation is slightly lower, but the deviation is increasing linearly. For the simulation with higher density the trend is similar, with a linear increase in deviation from the mass balance and the courser mesh having i slightly lower slope gradient than the finer mesh.

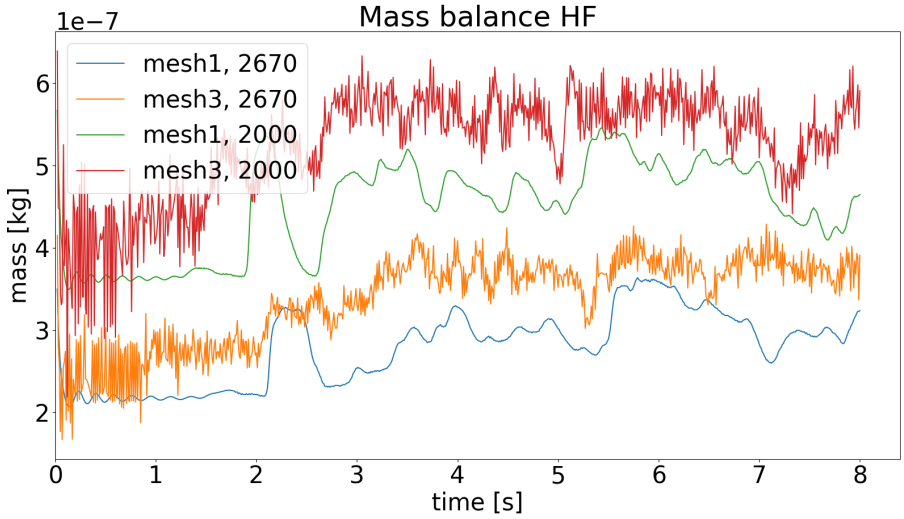


Figure 7.6: Shows the calculated mass balance for HF by calculating the difference in HF entering, leaving and what is accumulated in the system for two different solid phase density and different mesh.

The mass of the accumulated HF.particles in the system are shown in figure 7.7. For all simulations the mass of HF are increasing linearly.

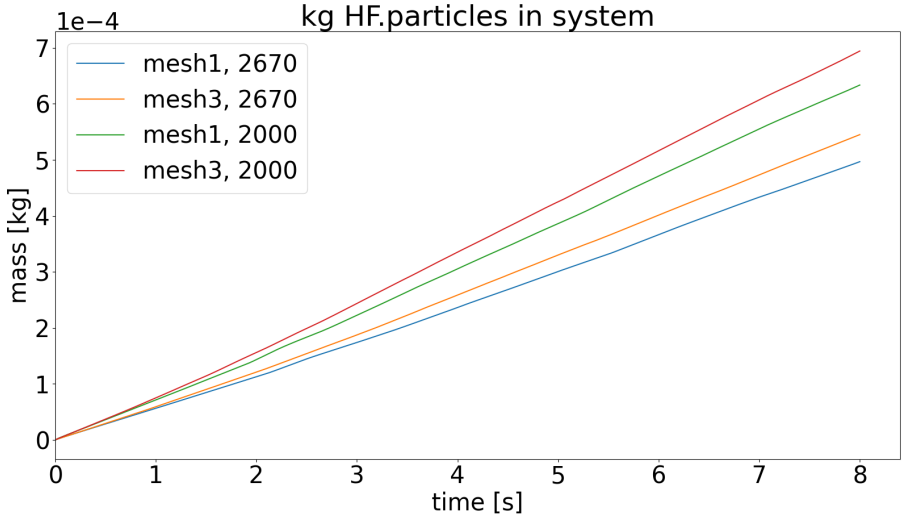


Figure 7.7: Shows the mass of the adsorbed HF in the system for two different solid phase density and different mesh.

The mass balance in figure 7.8 shows the mass balance, but the mass of the accumulated HF in the solid phase is not included. The reduced mass balance for all simulations are oscillating around the same negative value of -3.42·10 $^{-7}$.

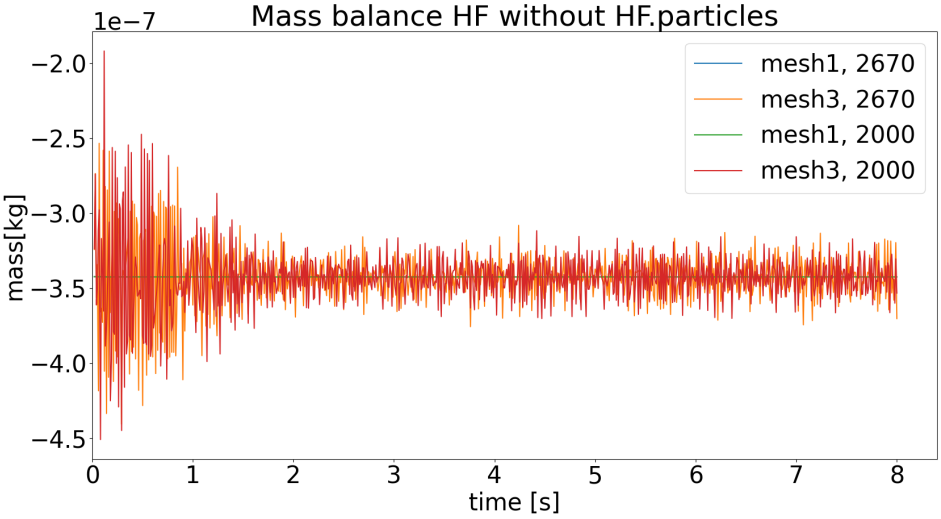
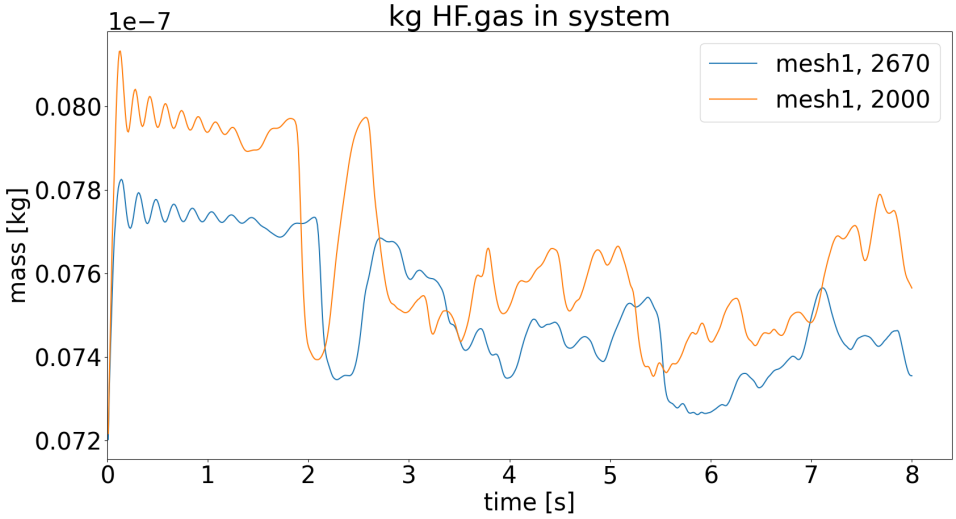
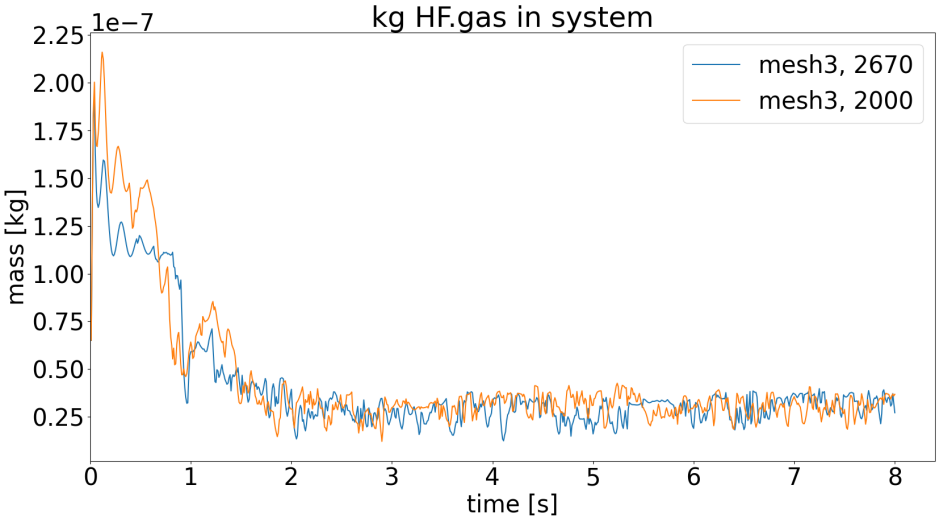


Figure 7.8: Shows the calculated mass balance of the system if the adsorbed HF is not included for two different solid phase density and different mesh.

The amount of HF.gas and HF_s.gas during the simulations are shown in figure 7.9 and 7.10. The courser mesh has a HF.gas and HF_s.gas concentration 2 order of magnitudes lower than the finer mesh. The difference in the concentration for the different meshes are the same.

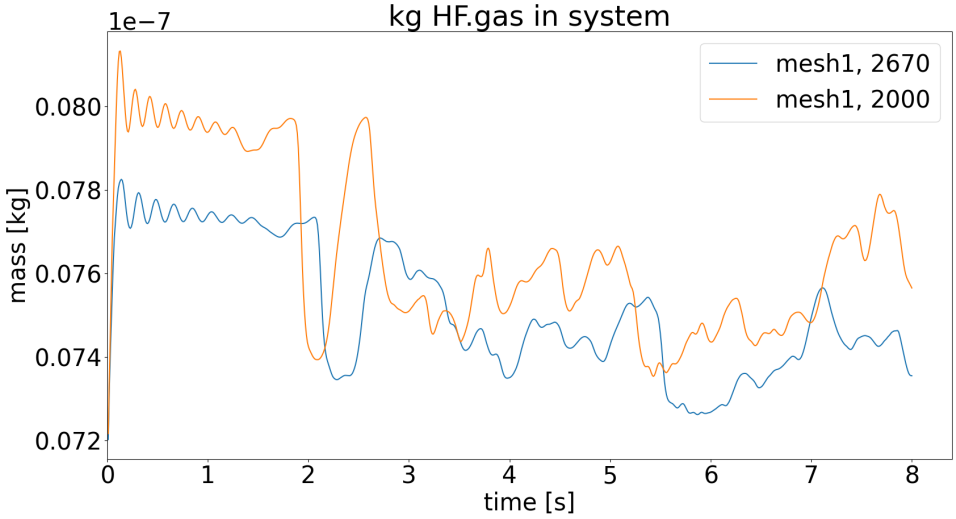


(a) mesh1

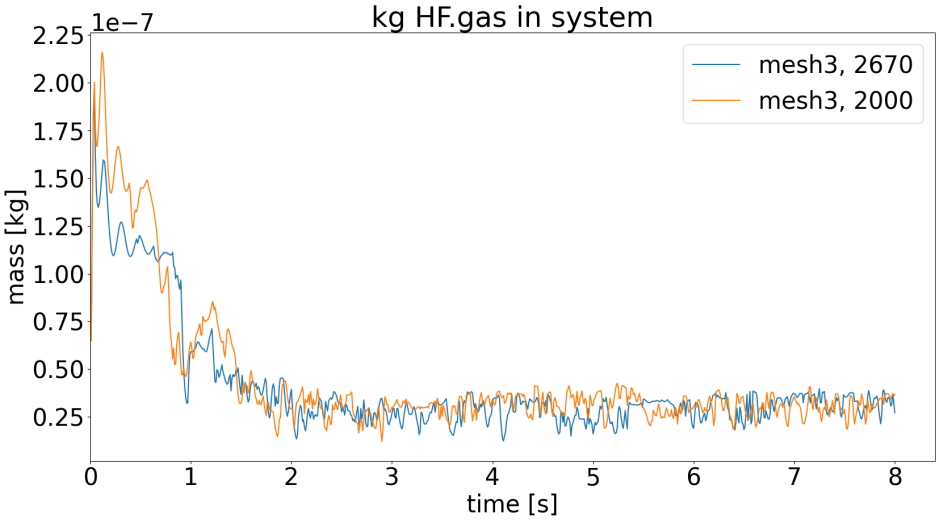


(b) mesh3

Figure 7.9: Shows the amount of HF.gas for two different solid phase density and different mesh as a function of time.



(a) mesh1



(b) mesh3

Figure 7.10: Shows the amount of HF_s.gas for two different solid phase density and different mesh as a function of time.

7.3 Case 3: Phase Transfer

Case 3 consists of three simulations. One simulation at 353 K, but with a low bed height, and two simulations with a higher bed height with temperatures at 353 K and 373K.

Figure 7.11 shows the amount of adsorbed HF divided by the adsorption amount of alumina in the system for the low bed at 353 K. The figure shows that the amount of adsorbed HF goes towards the adsorption capacity limit (red dotted line) of 4.3 mg/kg. Figure 7.12 shows the weight fraction composition on the outlet. The red line in the figure is the weight fraction on the inlet.

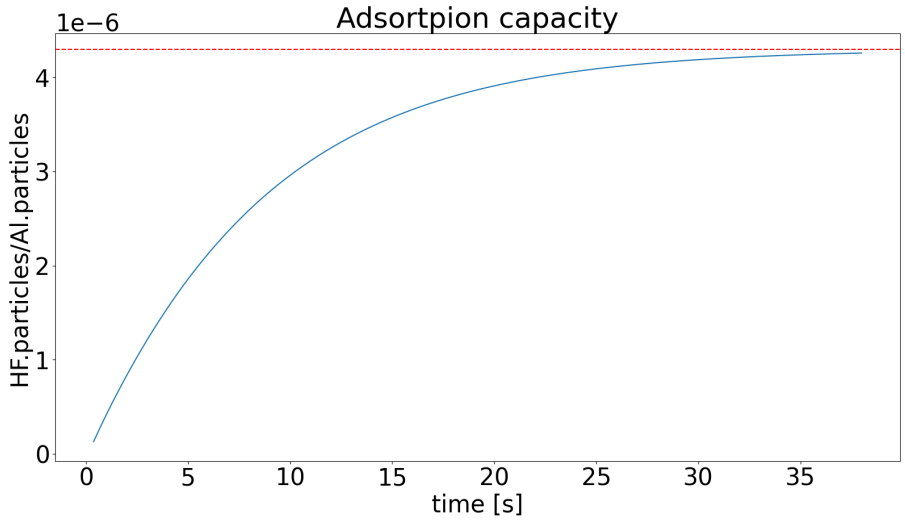


Figure 7.11: Shows the adsorption capacity limit as the red line and the amount of adsorbed HF in mg per kg alumina in the system for the simulation at 353 K and low bed height.

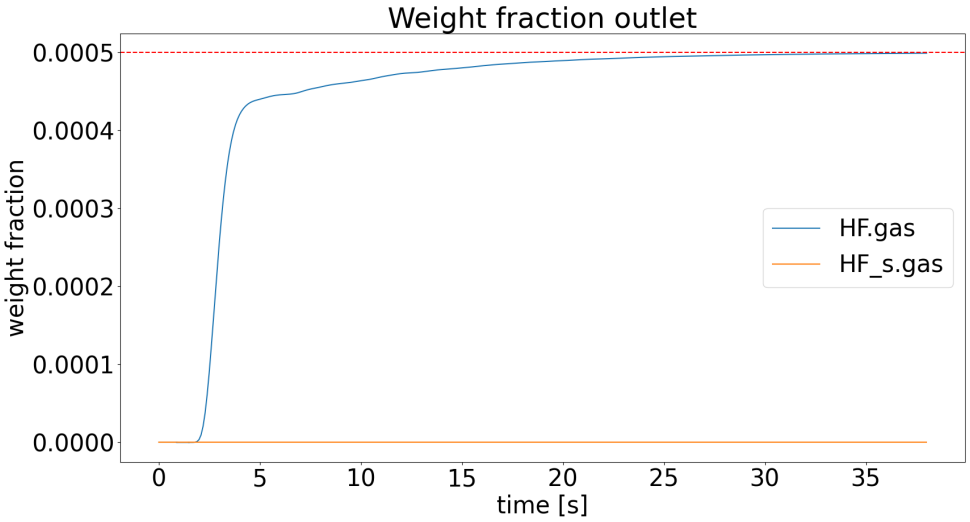


Figure 7.12: Shows the weight fraction of the gas at the outlet for the low bed height simulation at 353 K. The red line shows the weight fraction on the inlet.

Figure 7.13 shows the inlet mass flow into the system as a function of time for the two simulations with high bed height. The inlet flow rate is higher for the simulation with the lowest temperature. Figure 7.14 shows the mass of HF.particles in the system for the two simulations. The mass of adsorbed HF increases faster for the simulation at 373 K compared to the simulation at 353 K, but the simulation at lower temperature gets a higher total mass of adsorbed particles.

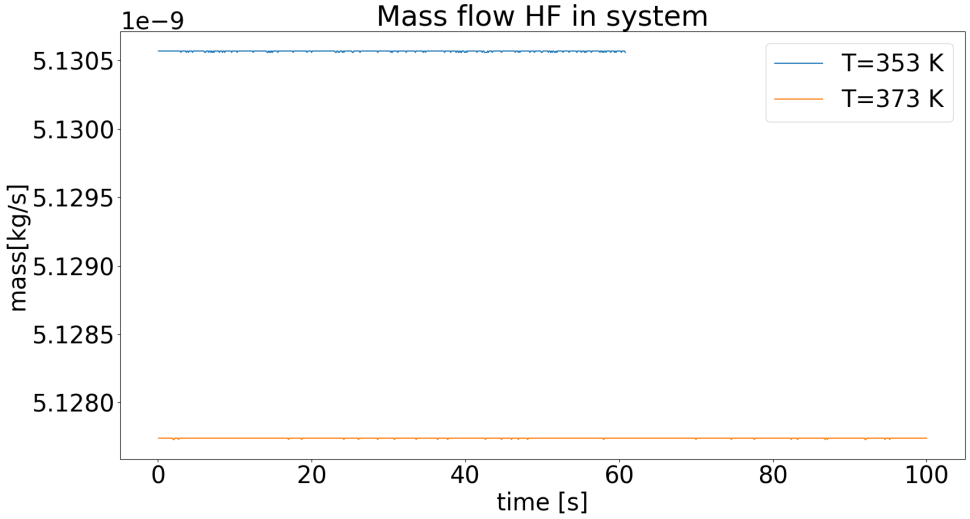


Figure 7.13: Shows the mass flow rate of HF in to the system for the simulations with increased bed heights at 353 and 373 K.

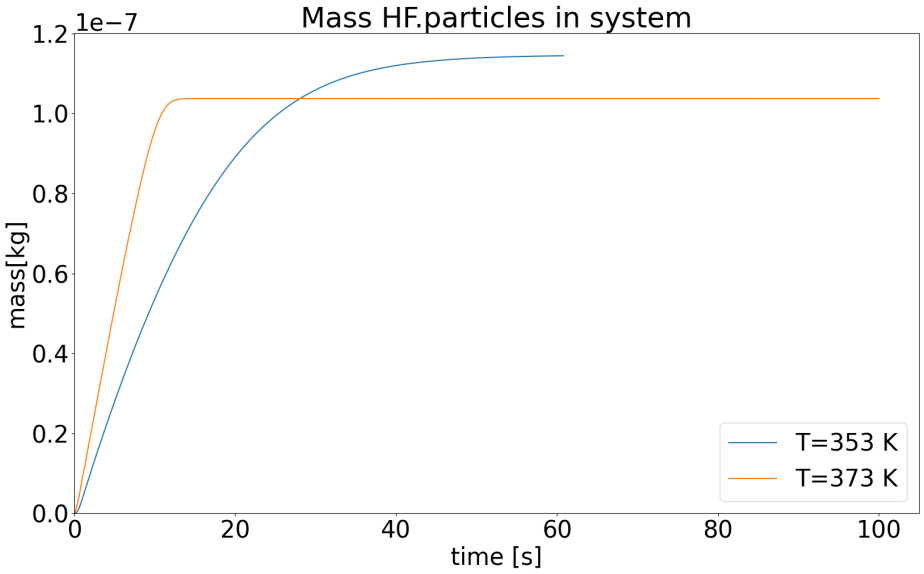


Figure 7.14: Shows the mass of adsorbed HF in the system as a function of time for the two simulations at.

Figure 7.15 shows the mass flow rate out of the system for the simulations at increased bed height at 353 and 373 K. The mass flow for the simulation at 353 K, increases fast at the beginning of the simulation. After the initial outlet mass flow rate increase, the flow rate increases slowly towards the inlet mass flow rate. The simulation at 373 K, has an initial lower mass flow rate of HF out of the system compared to the simulation at 353 K. After 10 seconds is the mass flow rate for the

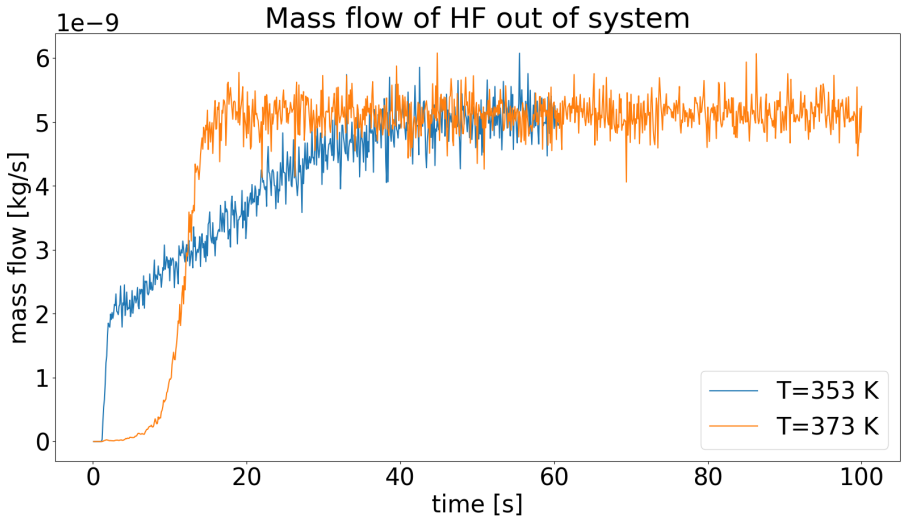


Figure 7.15: Shows the mass flow rate out of the system for simulations with increased bed height and temperature of 353 and 373 K

simulation at 373 K higher than the simulation at 353 K. Figure 7.16 shows the normalized adsorption capacity for the two simulations. The simulations at 373 K, have a faster increase in the fraction of used adsorption sites compared to the simulation at 353 K, and the fraction of used adsorption sites reaches 1 after 10 seconds. The simulation at 353 K reaches 1, after 50 seconds.

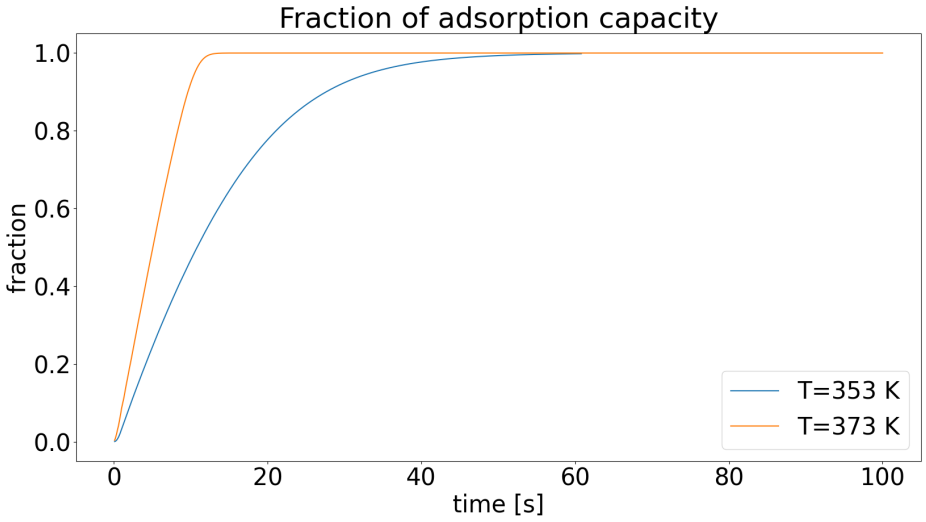


Figure 7.16: Shows the normalized fraction of available adsorption capacity used as a function of time for the the simulations with increased bed height and temperature of 353 and 373 K

8 Discussion

8.1 Gas cleaning of hydrogen fluoride

The use of HF in the alumina industry will most likely not cease in the near future, as it is essential to the production process Therefore the generation of HF will continue and so will the need for capturing it. The increase in operating temperature in the electrolytic cells, as well as the possible expansion of excising plants to meet the rising demand for aluminium, will put a strain on the GTC. Not only as increased gas volume, increased gas velocity, increased HF concentration, but the increase in temperature of the off-gas entering the GTC.

The implementation of heat exchangers is believed to be a good solution to cooling the off-gas, as it in addition can collect waste heat and possibly improves the flow profile inside the dry scrubber. With better technology and solution the implementation of heat exchangers will hopefully increase and the use of water cooling of the off-gas ceases, as water reduces the adsorption capacity [3]. The mechanism postulated by Sørhuus and Ose [27], concerning the regeneration of HF in the filter bags, has not been mentioned as a possible cause of increased emissions in the literature. Their paper presents no experiment or other evidence of regeneration taking place, but the mechanism should not be rejected on this basis. It should be possible to determine if the regeneration of HF in the filter bags is contributing to the increased emission by measuring the time between the increase in temperature and emissions. The effect of pore-blocking should not have a big impact on the emissions shortly after the increase in temperature or humidity, whereas the effect of regeneration should be possible to detect early.

New literature has an increased focus on pore-blocking, and it seems to have received more focus than surface area, especially in the work of Agbenyegah [3] has it received much focus. It is still not completely clear how the pores are blocked, but there is undeniable evidence for it to take place, and affecting the adsorption capacity and rate significantly. There is not a lot of literature that tries to quantify the adsorption rate, as most of the literature seems to be interested in the adsorption capacity, and how to maximize it. In this regard, the work of Agbenyegah [3], has been very useful as he presents multiple experiments, at different temperatures and concentrations, with a detailed explanation of his setup and parameters. It has proven difficult to find experiments in the literature that has included enough details to simulate their experiments. As well, most of the results in the literature are breakthrough plots that have little focus on the first hour of the adsorption. It was, therefore, difficult to find accurate and comparable data to validate a possible model.

The decrease in off-gas temperature entering the dry-scrubber seems to only have positive effects on the gas cleaning. The decrease in temperature is believed to increases the adsorption capacity due to reduction in reaction rate and pore blocking. There is a theoretical, optimal, adsorption temperature where the adsorption capacity is maximized without the reaction rate being affected. This temperature is likely a function of bed height, flow velocity, HF concentration and humidity levels. A good model of the adsorption reaction could be utilized to find this temperature.

8.2 The mathematical model

The mathematical model is not an exact representation of the real adsorption reaction. There are made some assumptions, meaning that some terms are not presented or considered. The system itself is not an accurate representation of a GTC, but the system can represent a possible down-scale part of a GTC found at some plants. The fluidized bed reactor was chosen because it is easier to model than an injection reactor, making the model development, simulation, and validation easier.

The system is assumed laminar, this makes the equation describing the system less complicated, and can possibly save some computational time.

The adsorption rate is modeled with Langmuir's kinetic equation, equation 49. There are proposed other models in the literature that can describe adsorption rates, but this is the most commonly used. The rewriting of the Arrhenius kinetic equation, and transforming it from a surface reaction rate to a volume-based reaction rate, should not impose any significant changes to the reaction rate or the accuracy of the model. The Langmuir adsorption kinetic is based on the adsorption isotherm, i.e. the adsorption capacity, as well as the chemical reaction rate coefficient. There have been conducted multiple studies investigating the adsorption capacity and attempted to determine the adsorption capacity. Previously the adsorption capacity has been closely linked to the surface area, but new research has found that pores are blocked, a large surface area might not mean a large adsorption capacity. There is a relation between surface area and adsorption capacity, but it not as straightforward as it was believed in earlier work, thus the adsorption capacity has proven difficult to determine.

In this work, the model bases the adsorption capacity on the regression model presented by Agbenyegah [3] that can be used to predict the fluoride loading at the point of breakthrough. The fluoride loading on the alumina particles at the breakthrough point is in this thesis called the kinetically available adsorption capacity. The thought behind this is to firstly have a measurement of the adsorption capacity, which is necessary to use the Langmuir kinetic equation. In addition, this is also the point when the reaction rate stops to be the time-dependent process, and mass transfer and inter-pore diffusion must be considered. There are multiple uncertainties in using the kinetically available adsorption capacity, as it first, is not know how accurate the regression model of Agbenyegah is. secondly, using the fluoride loading at breakthrough as a measurement for the adsorption capacity has not the author's knowledge been done before. This should highlight the need for verifying the model before use. In addition to how accurate the model should prove to be, it is easy to deduce that the accuracy of the model will decrease as the fluoride loading on the particles goes towards the kinetically available adsorption capacity. Another issue with this approach is that the regression model is only valid for the specific alumina particles used to generate it. The adsorption capacity, and therefore the kinetically available adsorption capacity will vary between alumina particles, dependent on the surface area, size of the alumina, and pore size distribution. It would therefore be necessary to determine the fluoride loading at breakthrough for different types of alumina used.

In addition to being based on the adsorption capacity, the Langmuir kinetic equation is based on the reaction rate coefficient. There were found no sources in literature stating the chemical reaction rate

of adsorption of HF, except for Agbenyegah [3] thermometric titration experiment. This method was highly inaccurate, in addition, it was only done one experiment at 100 ℃ and at one concentration. When calculating the reaction rate coefficient parameters in Arrhenius based on this value, it was impossible to determine how temperature-dependent the chemical reaction is. The activation energy was set at 150 kJ when it actually can be several times higher. In addition when calculating the reaction rate, only the exterior surface area is considered. The surface area due to the pores are included in the reaction rate coefficients.

8.3 Case1: No reaction

To investigate how the system behaved without phase transfer or reactions, five simulations were done to compare different mesh refinements and how the mesh affected the simulations. In addition, six simulates were conducted to investigate the particle volume fraction.

The bed heights were severely affected by the mesh, figure 7.1b, the bed height decreased when the cell size was reduced from 5.00 to 3.33 and 2.50 mm, figure 6.2b. The reduction in bed height can be explained by the formation and coalescence of bubbles. From figure 7.1b it is clear that in simulations with finer mesh, bubbles are formed. When the bubbles are formed, less gas is flowing between the particles. This is in accordance with the two-phase flow theory, equation 9. Comparing the simulations with figure 2.9, the bed behaves similarly. The simulation with the courses mesh (mesh 1) behaves like a smooth fluidized bed, while the simulations with mesh 3-5 behave like bubbling beds. The simulation with mesh 2 is a combination of both. This can indicate that the courser mesh doesn't capture essential effects in the system.

Comparing the mass of alumina and the volume of the solid phase of the different simulations it is clear from figures 7.2 and 7.3 that the courses mesh has the biggest deviation in mass and volume from the theoretical values. This is as expected as the distance between the cell centers is bigger for bigger cells, meaning that the distance of interpolated is longer. It is also clear from the figures that most of the inaccuracies in mass and particle-phase volume comes from the initial acceleration of the particles, i.e when the system goes from the initial stationary state to being accelerated by the gas that begins to flow into the system. This error is smaller for finer mesh than for courser, this can partly be because of the difference in cell size, but also a consequence of the time step for the simulations are being controlled by the Courant number. When the cells get smaller, the velocity is constant, the time step must be smaller when decreasing the Courant number. Simulation C1.7 has a lower maximum limit for the Courant number than C1.1, but they have the same volume increase, indicating that the time step is not the culprit behind the volume increase, but it is the cell size.

The biggest concern about the particle volume increase is the increase after the initial acceleration, after 1.5 seconds. The tolerance is small for both the alpha.particles and for the specie equation, so it is not believed to be iterative errors. This increase can possibly be because of badly defined boundary conditions. Changing the boundary conditions in simulation C1.10, so they match the boundary conditions used by Panicker et al. [71] did not solve the increase in particle volume. Simulation C1.0 without species, only defined phases, showed a close to constant particle volume

also the particle volume was closer to the theoretical volume. Simulation C1.9, simulation with no flow, appears constant after the first second, indicating that there might be a problem when the gas is flowing. This indicates an issue with the boundary conditions.

8.4 Case 2: Phase transfer, no adsorption limit

There is a limited amount of literature on the simulations of multiphase with multi-specie using multiphaseEulerFoam. No papers found used the reactionDriven phaseTransfer model. From the simulations in Case 1, two different densities for the solid phase were simulated with mesh 1 and 3. Since these simulations contain a phase transfer term, from gas to solid, and the solid phase has a constant density, the volume of the particle phase must increase, and is therefore not considered.

The inlet concentration of HF, in figure 7.5, are oscillating around $3.42{\cdot}10^{-7} kg$, this is in agreement with the theoretical mass flow of $3.42{\cdot}10^{-5} kg\ s^{-1}$ as the mass flow in figure 7.5 are plotted for every write time, that is set to be 0.01. This proves that the measurements and post-processing calculations are working as intended.

The mass balance of HF, shown in figure 7.6, shows that mass is not conserved. The mass balance of HF is wrong with the same order in magnitude as the mass flow into the system, and this is too much for the model to be accurate. Comparing the mass balance without the adsorbed HF, figure 7.8, with the mass flow into the system, figure 7.5, it is evident that most of the HF entering the system have been adsorbed, this is supported by figure 7.9 and figure 7.10 that shows that the amount of gaseous HF in the system, found as HF.gas and HF_s.gas, are low. The amount of gaseous HF and HF_s are relatively high at the beginning of the simulation, Herzog et al. [62] found that the bed reaches statistical steady state after 3 seconds. After 3 seconds the HF.gas and HF_s.gas are stable for the finer mesh. The difference in the amount of gaseous HF in the system for the different meshes can be related to the formation of bubbles in the simulation.

That the coarsest mesh, mesh1, shows a better mass balance and less oscillation. This is again a sign of the courser mesh being too coarse to give an accurate simulation of the system. The finer mesh has a better mass balance compared to the courser mesh. As discussed in Case 1 and shown in figure 7.3, the courser mesh has a larger increase in the particle volume compared to the finer mesh. If the mass of HF.particles was solely due to the increase in the particle volume the courser mesh should have had more HF.particles in the system than the finer mesh. This indicates that the main reason for the increased HF.particles mass is generated due to a programming error, or unintended bug. Where this error is in the code, is not easy to pinpoint.

Comparing the difference in HF.particles in the system for the different densities can help to pinpoint where the programming error is. The simulations with the highest density are producing less mass of HF.particles, indicating that the phase transfer term is divided by the phase density.

Even if the initial adsorption rate of gaseous HF is correct, the reaction rate is a function of the

amount of adsorbed HF, as is evident by the way the Langmuir kinetic equation is implemented, listing 5.2. Comparing the total mass balance with the mass balance without HF.particles, the mass of HF.particles in the system are roughly twice as much as it should be. It is possible that by multiplying the adsorption capacity by two, the model can give an indication of the adsorption reaction, but the generation of HF.particles will affect the system for longer simulation.

8.5 Case 3: Phase Transfer

The simulation with a lower bed height is simulated to investigate and verify the Langmuir adsorption kinetic code, listing 5.2. This is done by investigating the adsorption capacity, and the mass of adsorbed particles in the system. Figure 7.11 shows that the mass of adsorbed HF goes towards the adsorption capacity specified in the reaction file, table 6.11. To further support that the implemented Langmuir adsorption code works, the weight fraction of HF.gas on the outlet goes to the weight fraction of HF.gas at the inlet when the mass of adsorbed HF goes towards the adsorption limit.

For the simulations with increased bed heights, the mass flow rates are different. The increased mass flow rate for the simulation at lower temperatures is most likely a round of error, as even small variation in the calculations of the inlet velocity could lead to the difference seen in figure 7.13. The difference is relatively large, but they don't seem to have any significant effect on the mass of adsorbed HF, figure 7.14. The mass of the adsorbed HF in the two simulations is similar to what is to be expected. The simulation at 373 K shows a faster increase in mass HF.particles adsorbed compared to the simulation at 353 K. This is because the adsorption rate is too slow at the low temperature, as can be seen by the increased mass flow rate of HF.gas out of the system in figure 7.15. As the mass of adsorbed HF gets closer to the adsorption capacity, figure 7.16, the reaction rate decreases for both simulations. The simulations at 353 K has a higher adsorption capacity the adsorption rate is less affected by the adsorbed HF.

These simulations show how the model and simulations can be utilized to find the optimal temperature. The model needs more work so the mass balance is satisfied. The simulations at 353 and 373 K are using the calculated reaction rate that assumes the activation energy is 150 kJ. The low activation energy assumed, makes relatively small changes to the temperature have a relatively high impact on the reaction rate. If the activation energy is more than 150 kJ the adsorption rate between the simulations at 353 and 373 would be smaller and possibly negligible. It is possible that the activation energy is so high that a difference of 20-60 Kelvin don't change the chemical reaction rate. In this case, the difference in adsorption rate would be controlled by the adsorption capacity and the temperature should be decreased as much as possible to make the gas cleaning as effective as possible.

9 Conclusion

In this thesis, the literature concerning the adsorption reaction of hydrogen fluoride on alumina has been studied and presented. A model in multiphaseEulerFoam was created to simulate the adsorption process with phase transfer from gas to solid in OpenFOAM. Small alterations to excising codes were made to describe the Langmuir adsorption kinetics. The following conclusion was reached as a result of the work done:

- The adsorption of hydrogen fluoride is heavily affected by changes in temperature. An increase in temperature increases the reaction rate and promotes the rate of pore-blocking.
- The reaction rate has successfully been modified to describe the adsorption rate with the Langmuir adsorption kinetic equation.
- The complete model presented does not work as intended. The mass of HF is not conserved, as the phase transfer term generates more adsorbed HF compared to what should be possible according to the mass balance.
- The amount of HF.particles in the system is approximately double of what it should be. Taking this into concern the model can be used to approximate the effect temperature has on the adsorption reaction.
- The biggest uncertainty related to the reaction rate is the lack of literature on the topic. The reaction rate has been fitted to Arrhenius reaction rate to estimate the Arrhenius coefficient A by assuming the activation energy of the chemisorption of HF is 150 kJ/mol.

10 Future Work

The literature is scars compared to other subjects related to the aluminium industry, but there is an increase in publication related to HF, both towards the mechanism and the adsorption capacity. For future work on simulating the adsorption of fluoride on alumina, the following future work is proposed:

- Information on the reaction rate of the adsorption reaction is scarce, as well as the work on the activation energy of the chemical reaction. It is necessary to have more accurate data for the adsorption reaction.
- Research on the adsorption capacity for alumina, and find a mathematical model for it.

For future work on the model presented in this thesis, the following work is proposed:

- Solve the problem with unstability and increase in particle volume.
- Solve the phase transfer problem where it generates too much mass.
- Implement turbulence and particle-particle interaction.
- Small scale experiments to collect data and see how alumina behaves in a liquidized bed.
- Validation and verification of the model and assumptions.
- Implement additional adsorption reaction rate models to see what is most accurate for the adsorption process.
- Implement mass transfer resistance.
- Look at heat and heat transfer.
- Simulate with multiple particle diameters.
- Simulate an injection scrubber, where alumina is added and removed.

References

[1] "Mineral Commodity Summaries 2020". In: *Aluminum Statistics and Information* (2020). URL: https://www.usgs.gov/centers/nmic/aluminum-statistics-and-information (visited on 01/26/2021).

[2] H. U. Sverdrup, K. V. Ragnarsdottir, and D. Koca. "Aluminium for the future: Modelling the global production, market supply, demand, price and long term development of the global reserves". In: *Resources, Conservation and Recycling* 103 (2015), pp. 139–154.

[3] G. E. K. Agbenyegah. "Mechanism and Kinetics of Hydrogen Fluoride Capture with Smelter Grade Alumina". In: (2019).

[4] E. Christensen. "Numerical Modeling of Hydrogen Fluoride: Creation of an OpenFOAM solver describing the adsorption reaction on alumina particles with the Langmuir isotherm. Initial study and development". unpublished. 2020.

[5] R. Lumley. *Fundamentals of aluminium metallurgy: production, processing and applications*. Elsevier, 2010.

[6] N. Tjahyono et al. "Fluoride emissions management guide (FEMG) for aluminium smelters". In: *Light Metals 2011*. Springer, 2011, pp. 301–306.

[7] C. Haidouti, A. Chronopoulou, and J. Chronopoulos. "Effects of fluoride emissions from industry on the fluoride concentration of soils and vegetation". In: *Biochemical systematics and ecology* 21.2 (1993), pp. 195–208.

[8] M. N. Ahmad et al. "Hydrogen fluoride damage to vegetation from peri-urban brick kilns in Asia: A growing but unrecognised problem?" In: *Environmental pollution* 162 (2012), pp. 319–324.

[9] M. Cheng. "Atmospheric chemistry of hydrogen fluoride". In: *Journal of Atmospheric Chemistry* 75.1 (2018), pp. 1–16.

[10] Y. Yang et al. "Modelling HF generation: the role of ambient humidity". In: *Light Metals 2014*. Springer, 2014, pp. 641–646.

[11] Y. J. Yang et al. "Modelling HF generation in aluminium reduction cell". In: *Canadian Metallurgical Quarterly* 54.2 (2015), pp. 149–160.

[12] E. C. Patterson. "Fluoride emissions from aluminium electrolysis cells". PhD thesis. ResearchSpace@ Auckland, 2002.

[13] W. Haupin and H. Kvande. "Mathematical model of fluoride evolution from Hall-Héroult Cells". In: *Essential Readings in Light Metals*. Springer, 1993, pp. 903–909.

[14] I. Paulin, Č. Donik, and M. Jenko. "Mechanisms of HF bonding in dry scrubber in aluminium electrolysis". In: *Materiali in tehnologije* 43.4 (2009), pp. 189–193.

[15] L. K. Wang et al. "Wet and dry scrubbing". In: *Air Pollution Control Engineering*. Springer, 2004, pp. 197–305.

[16] D. P. Clos et al. “Design of a cooled fouling probe to investigate scaling mechanisms from the aluminium production off-gas”. In: *Heat Exchanger Fouling and Cleaning* (2017), pp. 261–264.

[17] A. Sørhuus et al. “Gas Treatment in the GE Pot Integrated ABART Modules (PIA)”. In: *Light Metals 2021*. Springer, 2021, pp. 485–490.

[18] S. Broek. “Update on SO 2 Scrubbing Applied in Primary Aluminium Smelters”. In: *Light Metals 2020*. Springer, 2020, pp. 766–776.

[19] S. Glöser-Chahoud and C. Scheil. *Background informal technical document on techniques to reduce emissions from aluminium production*. Accessed: 29-5-2020.

[20] Y. Joumani et al. “Latest Developments in GTC Design to Reduce Fluoride Emissions”. In: *Light Metals 2021: 50th Anniversary Edition*. Springer International Publishing. 2021, pp. 451–458.

[21] J. H. De Boer. “50 endothermic chemisorption and catalysis”. In: *Advances in Catalysis*. Vol. 9. Elsevier, 1957, pp. 472–480.

[22] W. D. Lamb. “A study of the equilibrium adsorption of hydrogen fluoride on smelter grade aluminas”. In: *Essential Readings in Light Metals*. Springer, 1978, pp. 879–888.

[23] M. Baverez and R. De Marco. “Adsorption of Hydrogen Fluoride on Certain Smelter-Grade Aluminas”. In: *JOM* 32.1 (1980), pp. 10–14.

[24] J. F. Coyne et al. “Factors influencing hydrogen fluoride adsorption on alumina”. In: *Light Metals* (1989), pp. 113–118.

[25] A. R. Gillespie. “Mechanistic studies of HF adsorption on alumina”. PhD thesis. ResearchSpace@ Auckland, 1997.

[26] N. R Dando. “Adsorption/entrainment of fluoride in smelting grade alumina: surface chemical speciation and adsorption mechanism”. In: *Light Metals* 2 (2005), pp. 133–139.

[27] A. Sørhuus and S. Ose. “Pot gas treatment at high gas temperatures”. In: *Light Metals 2017*. Springer, 2017, pp. 495–500.

[28] A. de Gromard et al. “Development on electrolytic cell gas cooling”. In: *Light Metals 2014*. Springer, 2014, pp. 623–628.

[29] H. A. Al Qassab et al. “HEX retrofit enables smelter capacity expansion”. In: *Light Metals 2012*. Springer, 2012, pp. 815–820.

[30] M. Bonnier et al. “Development of a System Basedon Water Atomization to Decrease, Prior to Treatment, the Temperature of the Gas Emitted from Aluminum Cells”. In: *Light Metals* (2007), pp. 193–197.

[31] L. M. Perander et al. “Towards redefining the alumina specifications sheet—the case of HF emissions”. In: *Light Metals 2011*. Springer, 2011, pp. 285–290.

[32] N. R. Dando and S. J. Lindsay. “The competitive adsorption of HF and SO 2 on smelter grade alumina”. In: *Light Metals 2016*. Springer, 2016, pp. 527–531.

[33] G. J. McIntosh et al. “Adsorptive capacity and evolution of the pore structure of alumina on reaction with gaseous hydrogen fluoride”. In: *Langmuir* 31.19 (2015), pp. 5387–5397.

[34] M. Naderi. “Surface Area: Brunauer–Emmett–Teller (BET)”. In: *Progress in filtration and separation*. Elsevier, 2015, pp. 585–608.

[35] R. Bardestani, G. S. Patience, and S. Kaliaguine. “Experimental methods in chemical engineering: specific surface area and pore size distribution measurements—BET, BJH, and DFT”. In: *The Canadian Journal of Chemical Engineering* 97.11 (2019), pp. 2781–2791.

[36] A. R. Gillespie, M. M. Hyland, and J. B. Metson. “Irreversible HF adsorption in the dry-scrubbing process”. In: *JOM* 51.5 (1999), pp. 30–32.

[37] A. R. Gillespie, M. M. Hyland, and J. B. Metson. “The surface chemistry of secondary alumina from the dry scrubbing process”. In: *Essential Readings in Light Metals*. Springer, 2016, pp. 956–961.

[38] R. Juang and M. Chen. “Application of the Elovich equation to the kinetics of metal sorption with solvent-impregnated resins”. In: *Industrial & Engineering Chemistry Research* 36.3 (1997), pp. 813–820.

[39] H. A. Taylor and N. Thon. “Kinetics of Chemisorption1”. In: *Journal of the American Chemical Society* 74.16 (1952), pp. 4169–4173.

[40] Y. Yang et al. “The Adsorption Behavior of Moisture on Smelter Grade Alumina during Transportation and Storage—for Primary Aluminum Production”. In: *Metals* 10.3 (2020), p. 325.

[41] O. E. Bjarno and G. Wedde. *Device and a method of cleaning an effluent gas from an aluminium production electrolytic cell*. US Patent 9,242,203. 2016.

[42] A. Sørhuus, S. Ose, and E. Holmefjord. “Improved Abart Gas Treatment and Alumina Handling at the Karmøy Technology Pilot”. In: *TMS Annual Meeting & Exhibition*. Springer. 2018, pp. 655–661.

[43] A. Sørhuus, S. Ose, and B. M. Nilsen. “Possible use of 25 MW thermal energy recovered from the potgas at Alba line 4”. In: *Light Metals 2015*. Springer, 2015, pp. 631–636.

[44] E. H. Bouhabila et al. “An innovative compact heat exchanger solution for aluminum off-gas cooling and heat recovery”. In: *Light Metals 2013*. Springer, 2016, pp. 793–797.

[45] M. Nikolaisen, V. T. Skjervold, and T. Andresen. “Evaluation of Heat Recovery Heat Exchanger Design Parameters for Heat-to-Power Conversion from Metallurgical Off-Gas”. In: *IIR Rankine Conference 2020*. IIR. 2020.

[46] D. Brough and H. Jouhara. “The aluminium industry: A review on state-of-the-art technologies, environmental impacts and possibilities for waste heat recovery”. In: *International Journal of Thermofluids* 1 (2020), p. 100007.

[47] A. Sørhuus and G. Wedde. “Pot gas heat recovery and emission control”. In: *Essential Readings in Light Metals*. Springer, 2009, pp. 987–992.

[48] N. R. Dando and S. J. Lindsay. “Hard gray scale”. In: *Essential Readings in Light Metals*. Springer, 2008, pp. 602–607.

[49] R. Cocco, S. B. R. Karri, T. Knowlton, et al. “Introduction to fluidization”. In: *Chem. Eng. Prog* 110.11 (2014), pp. 21–29.

[50] D. Nemec and J. Levec. "Flow through packed bed reactors: 1. Single-phase flow". In: *Chemical Engineering Science* 60.24 (2005), pp. 6947–6957.

[51] M. J. Baker and G.R. Tabor. "Computational analysis of transitional air flow through packed columns of spheres using the finite volume technique". In: *Computers & chemical engineering* 34.6 (2010), pp. 878–885.

[52] D. Geldart. "Types of gas fluidization". In: *Powder technology* 7.5 (1973), pp. 285–292.

[53] J. M. V. Millán. "Introduction. The classical Geldart's diagram and the new type of gas-fluidization behavior". In: *Fluidization of Fine Powders*. Springer, 2013, pp. 1–6.

[54] W. P. Adamczyk. "Application of the numerical techniques for modelling fluidization process within industrial scale boilers". In: *Archives of Computational Methods in Engineering* 24.4 (2017), pp. 669–702.

[55] D. Kunii and O. Levenspiel. *Fluidization engineering*. Butterworth-Heinemann, 1991.

[56] C. Dechsiri. "Particle transport in fluidized beds". In: *University Medical Center Groningen, University of Groningen* (2004).

[57] P. D. S. de Vasconcelos and A. L. A. Mesquita. "Minimum and full fluidization velocity for alumina used in the aluminum smelter". In: *International Journal of Engineering Business Management* 3 (2011), p. 23.

[58] P. Erhard et al. *Prandtl-essentials of fluid mechanics*. Vol. 158. Springer Science & Business Media, 2010.

[59] H. A. Jakobsen. "Chemical reactor modeling". In: *Multiphase Reactive Flows* (2008).

[60] C. T. Crowe. *Multiphase flow handbook*. Vol. 59. CRC press, 2005.

[61] Y. Liu and O. Hinrichsen. "CFD modeling of bubbling fluidized beds using OpenFOAM®: Model validation and comparison of TVD differencing schemes". In: *Computers & chemical engineering* 69 (2014), pp. 75–88.

[62] N. Herzog et al. "A comparative study of different CFD-codes for numerical simulation of gas–solid fluidized bed hydrodynamics". In: *Computers & Chemical Engineering* 39 (2012), pp. 41–46.

[63] F. Taghipour, N. Ellis, and C. Wong. "Experimental and computational study of gas–solid fluidized bed hydrodynamics". In: *Chemical engineering science* 60.24 (2005), pp. 6857–6867.

[64] L. Li, B. Li, and Z. Liu. "Modeling of spout-fluidized beds and investigation of drag closures using OpenFOAM". In: *Powder Technology* 305 (2017), pp. 364–376.

[65] G. A. Bokkers. "Multi-level modelling of the hydrodynamics in gas phase polymerisation reactors". In: (2005).

[66] C. M Venier et al. "Comparing ANSYS Fluent® and OpenFOAM® simulations of Geldart A, B and D bubbling fluidized bed hydrodynamics". In: *International Journal of Numerical Methods for Heat & Fluid Flow* (2019).

[67] P. C. Johnson and R. Jackson. "Frictional–collisional constitutive relations for granular materials, with application to plane shearing". In: *Journal of fluid Mechanics* 176 (1987), pp. 67–93.

[68] A. Passalacqua and R. O. Fox. "Implementation of an iterative solution procedure for multi-fluid gas–particle flow models on unstructured grids". In: *Powder technology* 213.1-3 (2011), pp. 174–187.

[69] M. W. Hlawitschka, S. Drefenstedt, and H. J. Bart. "Local analysis of CO2 chemisorption in a rectangular bubble column using a multiphase Euler-Euler CFD code". In: *Journal Chemical Engineering Processing Technological* 7 (2016), pp. 3–10.

[70] K. R. Dupre et al. "Investigation of computational upscaling of adsorption of SO 2 and CO 2 in fixed bed columns". In: *Adsorption* 25.4 (2019), pp. 773–782.

[71] N. S. Panicker et al. "COMPUTATIONAL MODELING AND SIMULATION OF ALUMINIUM SMELTING PROCESS USING OPENFOAM". In: (2021).

[72] L. Yang, M. J. Nieves-Remacha, and K. F. Jensen. "Simulations and analysis of multiphase transport and reaction in segmented flow microreactors". In: *Chemical Engineering Science* 169 (2017), pp. 106–116.

[73] Y. Haroun, D. Legendre, and L. Raynal. "Volume of fluid method for interfacial reactive mass transfer: application to stable liquid film". In: *Chemical Engineering Science* 65.10 (2010), pp. 2896–2909.

[74] J. H. Ferziger, m. Perić, and R. L. Street. *Computational methods for fluid dynamics*. Vol. 4. Springer, 2020.

[75] C. Fernandes et al. "Validation of the CFD-DPM solver DPMFoam in OpenFOAM® through analytical, numerical and experimental comparisons". In: *Granular Matter* 20.4 (2018), pp. 1–18.

[76] M. Chrigui. "Eulerian-Lagrangian approach for modeling and simulations of turbulent reactive multi-phase flows under gas turbine combustor conditions". PhD thesis. Technische Universität, 2005.

[77] C. Venier et al. "Development of a Conservative Numerical Solver for Gas-Particles Multifluid Systems Using Kinetic Theory of Granular Flow". In: *Mecánica Computacional* 33.7 (2014), pp. 473–497.

[78] S. Busch, E. Fonias, and J. Andric. *A twophaseEulerFoam tutorial*. 2015.

[79] N. I. Kolev. "Drag, lift, and virtual mass forces". In: *Multiphase Flow Dynamics 2*. Springer, 2011, pp. 31–85.

[80] M. Upadhyay et al. "An Assessment of Drag Models in Eulerian–Eulerian CFD Simulation of Gas–Solid Flow Hydrodynamics in Circulating Fluidized Bed Riser". In: *ChemEngineering* 4.2 (2020), p. 37.

[81] S. P. Pudasaini. "A fully analytical model for virtual mass force in mixture flows". In: *International Journal of Multiphase Flow* 113 (2019), pp. 142–152.

[82] Z. Yu et al. "Effect of virtual mass force on the mixed transport process in a multiphase rotodynamic pump". In: *Advances in Mechanical Engineering* 6 (2014), p. 958352.

[83] C. E. Brennen. "A Review of Added Mass and Fluid Inertial Forces." In: (1982).

[84] J. Burelbach et al. "Thermophoretic forces on a mesoscopic scale". In: *Soft matter* 14.36 (2018), pp. 7446–7454.

[85] J. Joshi and A. K. Nayak. *Advances of computational fluid Dynamics in nuclear reactor design and safety assessment*. Woodhead Publishing, 2019, pp. 21–238.

[86] F. A. Bombardelli, A. E. González, and Y. I. Niño. "Computation of the particle Basset force with a fractional-derivative approach". In: *Journal of Hydraulic Engineering* 134.10 (2008), pp. 1513–1520.

[87] C. Reichhardt and C. J. O. Reichhardt. "Dynamics of Magnus-dominated particle clusters, collisions, pinning, and ratchets". In: *Physical Review E* 101.6 (2020).

[88] C. Kleinstreuer and Y. Feng. "Computational analysis of non-spherical particle transport and deposition in shear flow with application to lung aerosol dynamics–a review." In: *Journal of biomechanical engineering* 135.2 (2013), pp. 021008–021008.

[89] A. Faghri and Y. Zhang. *Fundamentals of Multiphase Heat Transfer and Flow*. Springer, 2020.

[90] W. Plazinski, W. Rudzinski, and A. Plazinska. "Theoretical models of sorption kinetics including a surface reaction mechanism: a review". In: *Advances in colloid and interface science* 152.1-2 (2009), pp. 2–13.

[91] K. J. Laidler. *Physical chemistry*. eng. 4th ed. Boston: Houghton Mifflin, 2003. ISBN: 0618123415.

[92] Y. Liu and L. Shen. "From Langmuir kinetics to first-and second-order rate equations for adsorption". In: *Langmuir* 24.20 (2008), pp. 11625–11630.

[93] K. J. Laidler. "The development of the Arrhenius equation". In: *Journal of chemical Education* 61.6 (1984), p. 494.

[94] A. K. Galwey and M. E. Brown. "Application of the Arrhenius equation to solid state kinetics: can this be justified?" In: *Thermochimica Acta* 386.1 (2002), pp. 91–98.

[95] H. S. Fogler and M. N. Gürmen. *Elements of Chemical Reaction Engineering. 2006*. Vol. 4. Pearson Education, Inc, 2006, pp. 757–811.

[96] J. H. Ferziger, M. Perić, and R. L. Street. "Introduction to Numerical Methods". In: *Computational Methods for Fluid Dynamics*. Cham: Springer International Publishing, 2020, pp. 23–40. ISBN: 978-3-319-99693-6. DOI: 10.1007/978-3-319-99693-6_2. URL: https://doi.org/10.1007/978-3-319-99693-6_2.

[97] *Finite Volume Method: A Crash introduction*. Accessed: 06-07-2021. URL: http://www.wolfdynamics.com/wiki/fvm_crash_intro.pdf.

[98] *OpenFOAM guide/Finite volume method (OpenFOAM)*. URL: https://openfoamwiki.net/index.php/OpenFOAM_guide/Finite_volume_method_(OpenFOAM)#:~:text=OpenFOAM's%5C%20finite%5C%20volume%5C%20method,can%5C%20be%5C%20selected%5C%20at%5C%20runtime..

[99] F. Moukalled, L. Mangani, and M. Darwish. "The Discretization Process". In: *The Finite Volume Method in Computational Fluid Dynamics: An Advanced Introduction with OpenFOAM® and Matlab*. Cham: Springer International Publishing, 2016, pp. 85–101. ISBN: 978-3-319-16874-6. DOI: 10.1007/978-3-319-16874-6_4. URL: https://doi.org/10.1007/978-3-319-16874-6_4.

[100] J. H. Ferziger, M. Perić, and R. L. Street. *Computational methods for fluid dynamics*. Vol. 3. Springer, 2002, pp. 447–498.

[101] J. E. Wallevik, K. Krenzer, and J. Schwabe. *Numerical Errors in CFD and DEM Modeling*. Ed. by N. Roussel and A. Gram. Dordrecht: Springer Netherlands, 2014, pp. 99–124. ISBN: 978-94-017-8884-7. DOI: 10.1007/978-94-017-8884-7_4. URL: https://doi.org/10.1007/978-94-017-8884-7_4.

[102] N. Roussel and A. Gram. "Simulation of fresh concrete flow". In: *RILEM State-of-the-Art Reports* 15 (2014).

[103] L. Peng et al. "Possible user-dependent CFD predictions of transitional flow in building ventilation". In: *Building and Environment* 99 (2016), pp. 130–141.

[104] F. Moukalled, L. Mangani, M. Darwish, et al. *The finite volume method in computational fluid dynamics*. Vol. 113. Springer, 2016.

[105] *About OpenFOAM*. URL: https://cfd.direct/openfoam/about/.

[106] L. Schulze and C. Thorenz. "The multiphase capabilities of the CFD toolbox OpenFOAM for hydraulic engineering applications". In: *ICHE 2014. Proceedings of the 11th International Conference on Hydroscience & Engineering*. 2014, pp. 1007–1016.

[107] *Courant number*. URL: https://www.openfoam.com/documentation/guides/latest/doc/guide-fos-field-CourantNo.html.

[108] *Field under-relaxation*. URL: https://www.openfoam.com/documentation/guides/latest/doc/guide-solvers-under-relaxation.html.

[109] T. Holzmann. "Mathematics, numerics, derivations and OpenFOAM®". In: *Loeben, Germany: Holzmann CFD* (2016).

[110] P. T. Peeters. "CFD of multiphase pipe flow: A comparison of solvers". PhD thesis. 2016.

[111] D. S. Márquez. "An extended mixture model for the simultaneous treatment of short and long scale interfaces". PhD thesis. 2013.

[112] E. Roohi, M. Pendar, and A. Rahimi. "Simulation of three-dimensional cavitation behind a disk using various turbulence and mass transfer models". In: *Applied Mathematical Modelling* 40.1 (2016), pp. 542–564.

[113] C. Greenshields. *OpenFOAM 8 Released: OpenFOAM Foundation*. 2020. URL: https://openfoam.org/release/8/.

[114] H. Weller. *multiphaseEulerFoam: replacement for reactingMultiphaseEulerFoam*. 2020. URL: https://github.com/OpenFOAM/OpenFOAM-dev/commit/b832453b72ffcd77502c582db5008298e384

[115] *multiphaseEulerFoam.C File Reference*. URL: https://cpp.openfoam.org/dev/multiphaseEulerFoa8C.html.

[116] *MultiComponentPhaseModel.C File Reference*. URL: https://cpp.openfoam.org/v8/MultiComponentPhaseModel_8C.html.

[117] *multiphaseEulerFoam.C File Reference*. URL: https://cpp.openfoam.org/v8/multiphaseEulerFoam8C.html.

[118] *src/thermophysicalModels/specie/reaction/Reactions/Reaction/Reaction.C File Reference*. URL: https://cpp.openfoam.org/v8/Reaction_8C.html.

[119] *surfaceArrheniusReactionRate Class Reference*. URL: https://cpp.openfoam.org/v8/classFoam_1_1surfaceArrheniusReactionRate.html.

[120] *phaseTransferModel Class Reference*. URL: https://cpp.openfoam.org/v8/classFoam_1_1phaseTransferModel.html.

[121] Outotec. *HSC Chemistry 9*. Version 9.9.2.3. URL: https://www.hsc-chemistry.com/.

[122] J. C. Posey. *Viscosity of Gaseous Anhydrous Hydrogen Fluoride*. Tech. rep. Carbide and Carbon Chemicals Co.(K-25), 1953.

[123] URL: https://doc.comsol.com/5.5/doc/com.comsol.help.cfd/cfd_ug_fluidflow_high_mach.08.27.html.

[124] Y. Zhang et al. "Adsorption equilibrium and kinetics of the removal of ammoniacal nitrogen by zeolite X/activated carbon composite synthesized from elutrilithe". In: *Journal of Chemistry* 2017 (2017).

[125] H. K. Boparai, M. Joseph, and D. M. O'Carroll. "Kinetics and thermodynamics of cadmium ion removal by adsorption onto nano zerovalent iron particles". In: *Journal of hazardous materials* 186.1 (2011), pp. 458–465.

Appendices

A Calculations

A.1 Calculation of reaction rate

Agbenyegah [3] found an reaction rate of $2.5 \cdot 10^{-8}$ mol/sg.
Assuming 1g;

Reaction rate $= 2.5 \cdot 10^{-8}$ mol/s.
The reaction rate can be written as;
Reaction rate $= k_c[C]A_s$.
The concentration is 320 mg/Nm^3 at 25 ℃.
From the ideal gas law;. $\frac{V_1}{T_1} = \frac{V_2}{T_2}$ assuming constant pressure.
The concentration of HF at 100 ℃ will be 320/4 mg/m^3 = 80 $mg/m^3 = 4 \cdot 10^{-3}$ mol/m^3 when HF has a molar mass of 20.01 g/mol.

Alumina has a density of $3760 \cdot 10^3$ g/m^3, and by assuming no porosity 1g is equivalent of $2.660 \cdot 10^{-7}$ m^3.

Using the area/volume formula for a sphere to get the surface area of one gram when the particles have a mean diameter of $70 \cdot 10^{-6}$ m; the total surface area is

$$2.66 \cdot 10^{-7}\ m^3 \frac{3}{70 \cdot 10^{-6}\ m} = 1.14 \cdot 10^{-2}\ m^2$$

Then the reaction rate coefficient, k_c is calculated to;

$$\frac{Reaction\ rate}{[C]A_s} = k_c = \frac{2.5 \cdot 10^{-8}}{4 \cdot 10^{-3} \cdot 1.14 \cdot 10^{-2}} = 5.48 \cdot 10^{-4}$$

The reaction rate coefficient can be expressed by the Arrhenius equation;

$$k_c = A \cdot exp\left(-\frac{E_a}{TR}\right)$$

The activation energy for chemical adsorption is 40∼800 KJ/mol [124, 125]. There is done none to little research on the activation energy, so the activation energy is set to 150 kJ/mol. By assuming the activation energy, the pre-exponential factor, A can be calculated;

$$E_a = 1.5 \cdot 10^5 J/mol => A = \frac{5.48 \cdot 10^{-4}}{exp\left(-\frac{1.5 \cdot 10^5}{8.315 \cdot 373}\right)} = 5.55 \cdot 10^{17}$$

A.2 Flow rate, Case 3

Every measurements is given in sccm, assume the standard is 273,15 K and 1 atm (101325 Pa) pressure. Calculating the mass flow, since the volume of a gas is dependent on temperature. Assuming perfect gases;

Density of gas phase is assumed to be equal to that of pure Nitrogen;

$$\frac{0.001p}{RT_g} * (wt\%N_2 M_{N_2})kg/m^3 = \frac{0.001 \cdot 101325}{8.3144 \cdot 273.15} \cdot 28.014g/mol = 1.25kg/m^3$$

The mass flow of gas is then calculated by;

$$8.33 * 10^{-6}m^3/s \cdot 1.25kg/m^3 = 1.04 \cdot 10^{-5}kg/s = 1.04 \cdot 10^{-2}g/s$$

The HF gas concentration of interest is given at 650 mg/Nm^3. The volume flow is;

$$8.33 * 10^{-}6Nm^3/s \cdot 650mg/Nm^3 = 5.41 \cdot 10^{-3}mg/s = 5.41 \cdot 10^{-6}g/s$$

The weight fraction N_2 and HF is 0.9995 and 0.0005 respectfully.

Since the mass flow must be constant despite temperature, the inlet velocity is calculated from the mass flow.

For 353 K the gas inlet flow velocity is;

$$\frac{1.04 \cdot 10^{-5}kg/s}{\frac{0.001 \cdot 101325}{8.3144 \cdot 353.15} \cdot 28.014g/mol} \frac{1}{2.01 * 10^{-4}m^2} = 0.0535$$

For 373 K the gas inlet flow velocity is;

$$\frac{1.04 \cdot 10^{-}5kg/s}{\frac{0.001 \cdot 101325}{8.3144 \cdot 373.15} \cdot 28.014g/mol} 1 \frac{1}{2.01 * 10^{-4}m^2} = 0.0565$$

A.3 Calculation of Sutherland Coefficient for HF gas

From the graph in [122], two viscosity values were found;
$\mu(T = 300) = 1.25 \cdot 10^{-5}$
$\mu(T = 350) = 1.25 \cdot 10^{-5}$

In OpenFOAM the Sutherland's formula for viscosity is;

$$\mu = \frac{A_s T^{\frac{3}{2}}}{T + T_s}$$

Rewritten for A_s and T_s;

$$A_s = \frac{\mu(T + T_s)}{T^{\frac{3}{2}}}$$

$$T_s = \frac{A_s T^{\frac{3}{2}}}{\mu} - T$$

With two equations and two unknowns, inserting $A_s(T = 300)$ into $T_s(T = 350)$;

$$T_s = \frac{1.25 \cdot 10^{-5}(300 + T_s)}{300^{\frac{3}{2}}} \frac{350^{\frac{3}{2}}}{1.47 \cdot 10^{-5}} - 350$$

Calculations will yield;
$T_s = 402$
$A_s = 1.69 \cdot 10^{-6}$

B Post-Processing

B.1 Paraview

Figure B.1 shows the flow scheme with reaction and phase transformation for the system. Equation 65 shows the amount of HF accumulated in the system for each time step, note that there is no diffusion out of the system due to the boundary conditions. In the simulations HF.temp is called HF_s.

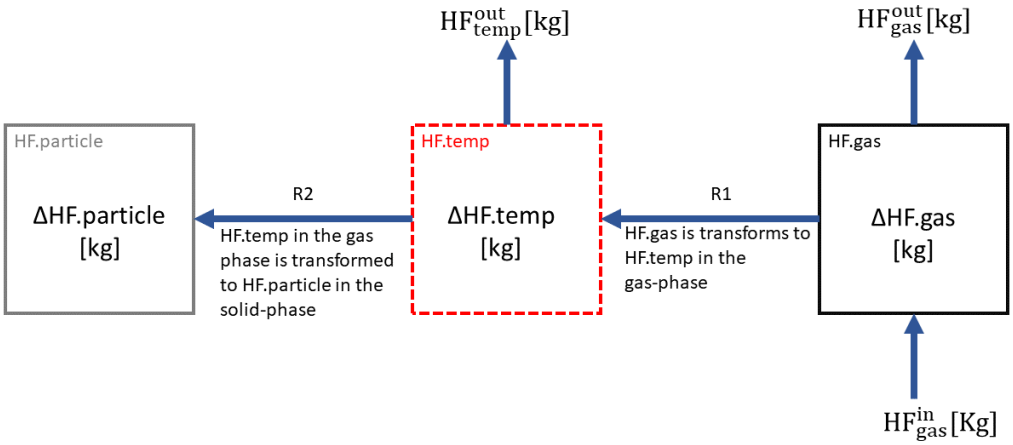


Figure B.1: Schematic of the adsorption reaction, flow and accumulation of different species of interest in the system

$$\Delta HF_{gas} = HF^{in}_{gas} - HF^{out}_{gas} + HF^{diff}_{gas} - R1 \tag{65}$$

Equation 66 calculates the accumulation of the intermediate gas phase in the system. This specie is dependent on the reaction rate (R1) of HF.gas to HF.temp, as well as the phase transformation rate (R2) of HF.temp to HF.particle.

$$\Delta HF_{temp} = R1 + HF^{in}_{temp} - HF^{out}_{temp} - R2 \tag{66}$$

Equation 67 calculates the accumulation of HF.particle, i.e. the phase transformation. This phase transformation is only dependent on the phase transformation rate, R2.

$$\Delta HF_{particle} = R2 \tag{67}$$

Combining equation 65, 66 and equation 67 to get the total mass balance, which should in theory be equal to zero:

$$HF_{gas}^{in} + HF_{gas}^{diff} - HF_{gas}^{out} - (\Delta HF_{temp} + HF_{temp}^{out} + \Delta HF_{particle} - HF_{temp}^{in}) - \Delta HF_{gas} = 0 \tag{68}$$

To calculate the theoretical mass flow of a specie s, in phase i with a total of n species, into the system the following equation, where the density of a phase is the sum of specie weight fraction multiplied by specie density, was used.

$$\dot{m}_s = \dot{V}_i Y_s \rho_i = \alpha_i A u_y Y_s \frac{0.001p}{RT_g} \cdot \sum_{s=1}^{n} (M_s \cdot Y_s) \tag{69}$$

where Y_s is the weight fraction of specie s in phase i, A is the inlet area, u_y is the velocity component normal to the inlet, p is the pressure and M_s is the molar mass of specie s.

To calculate the mass of specie s in phase i with n species, equation 70 and 71 was used for the solid and compressible gas phase, respectively.

$$m_s^s = V_s \rho_s = \int_V \alpha_i Y_s \sum_{i=1}^{n} (Y_s \rho_s) dV \tag{70}$$

$$m_s^g = V_s \rho_s = \int_V \alpha_i Y_s \frac{0.001p}{RT_g} \cdot \sum_{i=1}^{n} (M_s \cdot Y_s) dV \tag{71}$$

To get the mass in the system post-processing in Paraview version 5.6.2 was used, the post-processing workflow is shown in figure B.2, where the equation in the calculators are equation 70 or 71 depending on the specie of interest.

In preview the following codes were used:
mass HF_s.particles:

$HF_s.particles * alpha.particles * 2670$

mass of Al2O3.particles

$Al2O3.particles * alpha.particles * 2670$

An example for the mass of gas in the system consisting of (HF.gas, HF_s.gas and N2.gas):

$alpha.gas * HF.gas * (0.001 * p/(8.3144 * T.gas)) * (20.01 * HF.gas + 28.01 * N2.gas + 20.01 * HF_s.gas)$

For the other gas calculations the first HF.gas term is substituted with the gas field. To calculate the mass flow rate of a gas over a patch, the following code was used:

$alpha.gas * HF.gas * (0.001 * p/(8.3144 * T.gas)) * (20.01 * HF.gas + 28.01 * N2.gas + 20.01 * HF_s.gas) * U$

The complete path in paraview for on calculation is shown in figure B.2

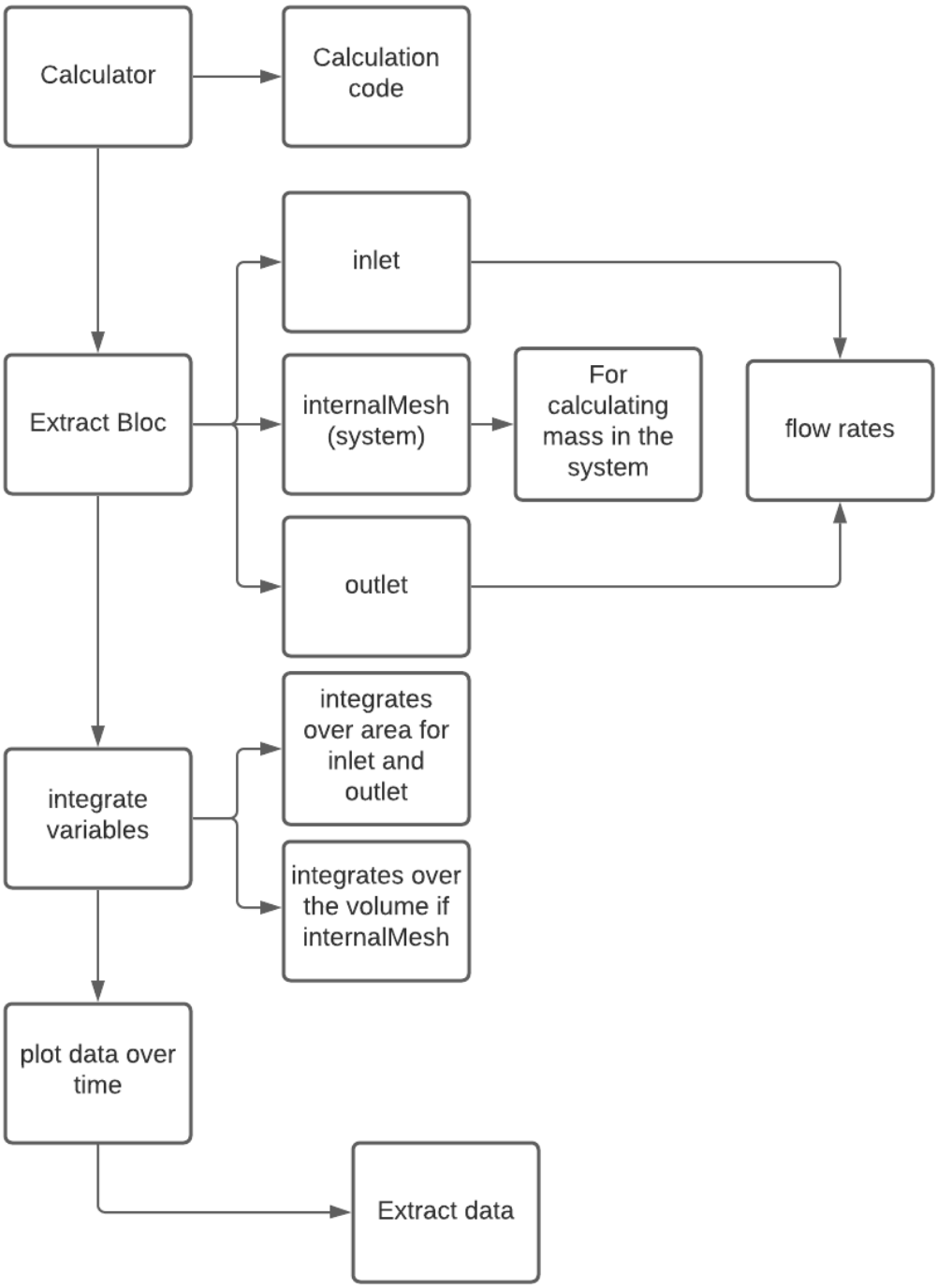


Figure B.2: Flow chart of post processing in Paraview

C Code: AdsorptionArrhenius

C.1 adsorptionArrheniusReactionRateI.H

```
/*---------------------------------------------------------------------------*\
  =========                 |
  \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox
   \\    /   O peration     | Website:  https://openfoam.org
    \\  /    A nd           | Copyright (C) 2019 OpenFOAM Foundation
     \\/     M anipulation  |
-------------------------------------------------------------------------------
License
    This file is part of OpenFOAM.

    OpenFOAM is free software: you can redistribute it and/or modify it
    under the terms of the GNU General Public License as published by
    the Free Software Foundation, either version 3 of the License, or
    (at your option) any later version.

    OpenFOAM is distributed in the hope that it will be useful, but WITHOUT
    ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or
    FITNESS FOR A PARTICULAR PURPOSE.  See the GNU General Public License
    for more details.

    You should have received a copy of the GNU General Public License
    along with OpenFOAM.  If not, see <http://www.gnu.org/licenses/>.

  adsorptionArrheniusReactionRateI.H
\*---------------------------------------------------------------------------*/
#include "volFields.H"
#include "mathematicalConstants.H"
#include "thermodynamicConstants.H"

using namespace Foam::constant::mathematical;
using namespace Foam::constant::thermodynamic;
// * * * * * * * * * * * * * * * * * * Constructors  * * * * * * * * * * * * * * * * * //

inline Foam::adsorptionArrheniusReactionRate::adsorptionArrheniusReactionRate
(
    const speciesTable& st,
    const objectRegistry& ob,
    const dictionary& dict
)
:
    ArrheniusReactionRate(st, dict),
    aName_(dict.lookup("a")),
  aField_
    (
        ob.lookupObject<volScalarField::Internal>(aName_)
    ),
  adsMax_(dict.lookupOrDefault<scalar>("adsMax",1)),
  adsoName_(dict.lookup("adso")),
  adsoField_
```

```
50      (
51          ob.lookupObject<volScalarField>(adsoName_)
52      ),
53    adsorbentName_(dict.lookup("adsorbent")),
54    adsorbentField_
55      (
56          ob.lookupObject<volScalarField>(adsorbentName_)
57      )
58 {}
59
60
61 // * * * * * * * * * * * * * * * * * * Member Functions  * * * * * * * * * * * * * * * //
62
63 inline Foam::scalar Foam::adsorptionArrheniusReactionRate::operator()
64 (
65     const scalar p,
66     const scalar T,
67     const scalarField& c,
68     const label li
69 ) const
70 {
71
72   scalar M = 1 - (((adsoField_[li]*pow(10,-6))/(adsorbentField_[li]+VSMALL))/adsMax_)
       ;
73
74     return ArrheniusReactionRate::operator()(p, T, c, li)*aField_[li]*(M);
75 }
76
77 inline Foam::scalar Foam::adsorptionArrheniusReactionRate::ddT
78 (
79     const scalar p,
80     const scalar T,
81     const scalarField& c,
82     const label li
83 ) const
84 {
85   //scalar M = 1 - (((adsoField_[li]*1e6)/(adsorbentField_[li]))/adsMax_);
86   scalar M = 1 - (((adsoField_[li]*pow(10,-6))/(adsorbentField_[li]+VSMALL))/adsMax_)
       ;
87     return ArrheniusReactionRate::ddT(p, T, c, li)*aField_[li]*(M);
88 }
89
90
91 inline void Foam::adsorptionArrheniusReactionRate::write(Ostream& os) const
92 {
93     ArrheniusReactionRate::write(os);
94     writeEntry(os, "a", aName_);
95   writeEntry(os, "adsMax", adsMax_);
96 }
97
98
99 inline Foam::Ostream& Foam::operator<<
100 (
101     Ostream& os,
```

```
102     const adsorptionArrheniusReactionRate& arr
103 )
104 {
105     arr.write(os);
106     return os;
107 }
108
109 // ************************************************************************* //
```

C.2 adsorptionArrheniusReactionRate.H

```
/*---------------------------------------------------------------------------*\
  =========                 |
  \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox
   \\    /   O peration     | Website:  https://openfoam.org
    \\  /    A nd           | Copyright (C) 2019 OpenFOAM Foundation
     \\/     M anipulation  |
-------------------------------------------------------------------------------
License
    This file is part of OpenFOAM.

    OpenFOAM is free software: you can redistribute it and/or modify it
    under the terms of the GNU General Public License as published by
    the Free Software Foundation, either version 3 of the License, or
    (at your option) any later version.

    OpenFOAM is distributed in the hope that it will be useful, but WITHOUT
    ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or
    FITNESS FOR A PARTICULAR PURPOSE.  See the GNU General Public License
    for more details.

    You should have received a copy of the GNU General Public License
    along with OpenFOAM.  If not, see <http://www.gnu.org/licenses/>.

Class
    Foam::adsorptionArrheniusReactionRate

Description
    A modified Arrhenius reaction rate that calculates the reaction rate, and
    including the
  consertration of empty adsorption cites (M):

        k = (A * T^beta * exp(-Ta/T))*a*M

    Where a is the surface area per unit volume, the name of which is
    specified by the user.
  M is calculated, from the user defined adsorption maximum (adsMax [mg/Kg]) and
  the user given adsorbed specie, as [Maximum]-[Adsorbed].

SourceFiles
    adsorptionArrheniusReactionRateI.H

  adsorptionArrheniusReactionRate.H

\*---------------------------------------------------------------------------*/

#ifndef adsorptionArrheniusReactionRate_H
#define adsorptionArrheniusReactionRate_H

#include "ArrheniusReactionRate.H"
#include "volFields.H"
#include "scalarField.H"
#include "typeInfo.H"
```

```
52
53 // * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //
54
55 namespace Foam
56 {
57
58 // Forward declaration of friend functions and operators
59
60 class adsorptionArrheniusReactionRate;
61
62 Ostream& operator<<(Ostream&, const adsorptionArrheniusReactionRate&);
63
64 /*---------------------------------------------------------------------------*\
65                Class adsorptionArrheniusReactionRate Declaration
66 \*---------------------------------------------------------------------------*/
67
68 class adsorptionArrheniusReactionRate
69 :
70     public ArrheniusReactionRate
71 {
72     // Private Data
73
74         //- Name of the surface area per unit volume field
75         const word aName_;
76
77         //- The surface area per unit volume field
78         const scalarField& aField_;
79
80     //- Adsorption capacity
81     const scalar adsMax_;
82
83     //- Name of adsorbed specie
84     const word adsoName_;
85
86     //- The surface area per unit volume field
87         const scalarField& adsoField_;
88
89     //- Name of adsorbent specie
90     const word adsorbentName_;
91
92     //- The surface area per unit volume field
93         const scalarField& adsorbentField_;
94
95 public:
96
97     // Constructors
98
99         //- Construct from dictionary
100        inline adsorptionArrheniusReactionRate
101        (
102            const speciesTable& species,
103            const objectRegistry& ob,
104            const dictionary& dict
105        );
```

```
106
107     // Member Functions
108
109         //- Return the type name
110         static word type()
111         {
112             return "adsorptionArrhenius";
113         }
114
115         //- Evaluate the rate
116         inline scalar operator()
117         (
118             const scalar p,
119             const scalar T,
120             const scalarField& c,
121             const label li
122         ) const;
123
124         //- Evaluate the derivative
125         inline scalar ddT
126         (
127             const scalar p,
128             const scalar T,
129             const scalarField& c,
130             const label li
131         ) const;
132
133         //- Write to stream
134         inline void write(Ostream& os) const;
135
136     // Ostream Operator
137
138         inline friend Ostream& operator<<
139         (
140             Ostream&,
141             const adsorptionArrheniusReactionRate&
142         );
143 };
144
145 // * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //
146
147 } // End namespace Foam
148
149 // * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //
150
151 #include "adsorptionArrheniusReactionRateI.H"
152
153 // * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //
154
155 #endif
156
157 // ************************************************************************* //
```

C.3 makeadsorptionArrheniusReactions.C

```
/*---------------------------------------------------------------------------*\
  =========                 |
  \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox
   \\    /   O peration     | Website:  https://openfoam.org
    \\  /    A nd           | Copyright (C) 2019-2020 OpenFOAM Foundation
     \\/     M anipulation  |
-------------------------------------------------------------------------------
License
    This file is part of OpenFOAM.

    OpenFOAM is free software: you can redistribute it and/or modify it
    under the terms of the GNU General Public License as published by
    the Free Software Foundation, either version 3 of the License, or
    (at your option) any later version.

    OpenFOAM is distributed in the hope that it will be useful, but WITHOUT
    ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or
    FITNESS FOR A PARTICULAR PURPOSE.  See the GNU General Public License
    for more details.

    You should have received a copy of the GNU General Public License
    along with OpenFOAM.  If not, see <http://www.gnu.org/licenses/>.

\*---------------------------------------------------------------------------*/

#include "makeReaction.H"

#include "adsorptionArrheniusReactionRate.H"

#include "forCommonGases.H"

// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

namespace Foam
{
    forCommonGases
    (
        makeGeneralReaction,
        IrreversibleReaction,
        adsorptionArrheniusReactionRate
    )
}

// ************************************************************************* //
```

C.4 Make/files

```
makeadsorptionArrheniusReactions.C

LIB = $(FOAM_USER_LIBBIN)/libmyAdsorptionArrheniusReactions
```

C.5 Make/options

```
EXE_INC = \
    -I$(LIB_SRC)/thermophysicalModels/reactionThermo/lnInclude \
    -I$(LIB_SRC)/thermophysicalModels/basic/lnInclude \
    -I$(LIB_SRC)/thermophysicalModels/specie/lnInclude \
    -I$(LIB_SRC)/thermophysicalModels/functions/Polynomial \
    -I$(LIB_SRC)/ODE/lnInclude \
    -I$(LIB_SRC)/finiteVolume/lnInclude \
    -I$(LIB_SRC)/meshTools/lnInclude \
  -I$(LIB_SRC)/thermophysicalModels/chemistryModel/lnInclude

LIB_LIBS = \
    -lfluidThermophysicalModels \
    -lreactionThermophysicalModels \
    -lspecie \
    -lODE \
    -lfiniteVolume \
    -lmeshTools

```

D Code: myReactionDriven

D.1 myReactionDriven.C

```
/*---------------------------------------------------------------------------*\
  =========                 |
  \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox
   \\    /   O peration     | Website:  https://openfoam.org
    \\  /    A nd           | Copyright (C) 2019-2020 OpenFOAM Foundation
     \\/     M anipulation  |
-------------------------------------------------------------------------------
License
    This file is part of OpenFOAM.

    OpenFOAM is free software: you can redistribute it and/or modify it
    under the terms of the GNU General Public License as published by
    the Free Software Foundation, either version 3 of the License, or
    (at your option) any later version.

    OpenFOAM is distributed in the hope that it will be useful, but WITHOUT
    ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or
    FITNESS FOR A PARTICULAR PURPOSE.  See the GNU General Public License
    for more details.

    You should have received a copy of the GNU General Public License
    along with OpenFOAM.  If not, see <http://www.gnu.org/licenses/>.

\*---------------------------------------------------------------------------*/

#include "myReactionDriven.H"
#include "phasePair.H"
#include "phaseSystem.H"
#include "addToRunTimeSelectionTable.H"

// * * * * * * * * * * * * * * * * Static Data Members * * * * * * * * * * * * * * * //

namespace Foam
{
namespace phaseTransferModels
{
    defineTypeNameAndDebug(myReactionDriven, 0);
    addToRunTimeSelectionTable(phaseTransferModel, myReactionDriven, dictionary);
}
}

// * * * * * * * * * * * * * * * * * * Constructors  * * * * * * * * * * * * * * * * * //

Foam::phaseTransferModels::myReactionDriven::myReactionDriven
(
    const dictionary& dict,
    const phasePair& pair
)
:
```

```
50      phaseTransferModel(dict, pair),
51      reactingName_(dict.lookup("reactingPhase")),
52      reactingPhase_
53      (
54          reactingName_ == pair_.first() ? pair_.phase1() : pair_.phase2()
55      ),
56      otherPhase_
57      (
58          pair.otherPhase(reactingPhase_)
59      ),
60      sign_
61      (
62          reactingName_ == pair_.first() ? -1 : 1
63      ),
64      species_(dict.lookup("species"))
65  {}
66
67  // * * * * * * * * * * * * * * * * * Destructor  * * * * * * * * * * * * * * * //
68
69  Foam::phaseTransferModels::myReactionDriven::~myReactionDriven()
70  {}
71
72  // * * * * * * * * * * * * * * * * * Member Functions  * * * * * * * * * * * * * //
73
74  const Foam::hashedWordList&
75  Foam::phaseTransferModels::myReactionDriven::species() const
76  {
77      return species_;
78  }
79
80  Foam::HashPtrTable<Foam::volScalarField>
81  Foam::phaseTransferModels::myReactionDriven::dmidtf() const
82  {
83      HashPtrTable<volScalarField> result;
84
85      forAll(species_, i)
86      {
87          const word name = species_[i];
88
89          volScalarField& Y =
90              const_cast<volScalarField&>(reactingPhase_.Y(name));
91
92          result.set
93          (
94              species_[i],
95              (sign_*reactingPhase_.R(Y) & Y).ptr()
96          );
97      }
98
99      return result;
100 };
101
102 // ************************************************************************* //
```

D.2 myReactionDriven.H

```
/*---------------------------------------------------------------------------*\
  =========                 |
  \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox
   \\    /   O peration     | Website:  https://openfoam.org
    \\  /    A nd           | Copyright (C) 2019-2020 OpenFOAM Foundation
     \\/     M anipulation  |
-------------------------------------------------------------------------------
License
    This file is part of OpenFOAM.

    OpenFOAM is free software: you can redistribute it and/or modify it
    under the terms of the GNU General Public License as published by
    the Free Software Foundation, either version 3 of the License, or
    (at your option) any later version.

    OpenFOAM is distributed in the hope that it will be useful, but WITHOUT
    ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or
    FITNESS FOR A PARTICULAR PURPOSE.  See the GNU General Public License
    for more details.

    You should have received a copy of the GNU General Public License
    along with OpenFOAM.  If not, see <http://www.gnu.org/licenses/>.

Class
    Foam::phaseTransferModels::reactionDriven

Description
    Phase transfer model representing change from one phase to another due
    to reactions. Intended for irreversible reactions.

SourceFiles
    reactionDriven.C

\*---------------------------------------------------------------------------*/

#ifndef myReactionDriven_H
#define myReactionDriven_H

#include "phaseTransferModel.H"

// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

namespace Foam
{

class phasePair;
class phaseModel;

namespace phaseTransferModels
{

/*---------------------------------------------------------------------------*\
```

```
53                        Class myReactionDriven Declaration
54 \*---------------------------------------------------------------------------*/
55
56 class myReactionDriven
57 :
58     public phaseTransferModel
59 {
60 private:
61
62     // Private Data
63
64         //- The name of the phase where the reactions occur
65         const word reactingName_;
66
67         //- Const reference to the reacting phase
68         const phaseModel& reactingPhase_;
69
70         //- Const reference to the other phase
71         const phaseModel& otherPhase_;
72
73         //- Sign used to multiply the source terms
74         const scalar sign_;
75
76         //- List of species changing phase
77         const hashedWordList species_;
78
79
80 public:
81
82     //- Runtime type information
83     TypeName("myReactionDriven");
84
85
86     // Constructors
87
88         //- Construct from components
89         myReactionDriven
90         (
91             const dictionary& dict,
92             const phasePair& pair
93         );
94
95
96     //- Destructor
97     virtual ~myReactionDriven();
98
99
100    // Member Functions
101
102        //- The list of individual species that are transferred
103        virtual const hashedWordList& species() const;
104
105        //- The mass transfer rate for individual species
106        virtual HashPtrTable<volScalarField> dmidtf() const;
```

```
};


// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

} // End namespace phaseTransferModels
} // End namespace Foam

// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

#endif

// ************************************************************************* //
```

D.3 Make/files

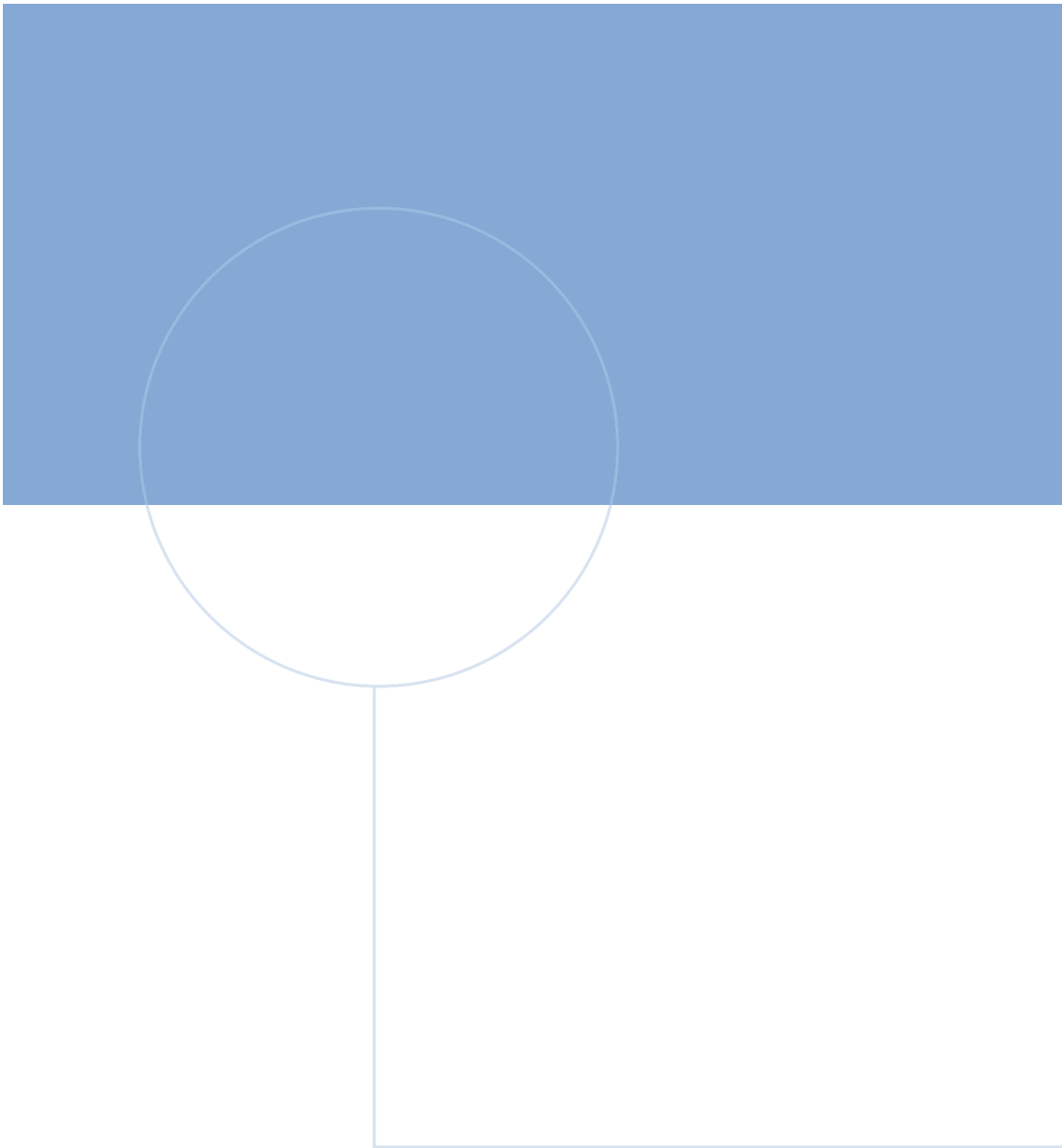
```
myReactionDriven.C

LIB = $(FOAM_USER_LIBBIN)/libmyReactionDriven
```

D.4 Make/options

```
EXE_INC = \
    -I$(FOAM_SOLVERS)/multiphase/multiphaseEulerFoam/interfacialModels/lnInclude \
    -I$(FOAM_SOLVERS)/multiphase/multiphaseEulerFoam/phaseSystems/lnInclude \
    -I$(LIB_SRC)/thermophysicalModels/basic/lnInclude \
    -I$(LIB_SRC)/MomentumTransportModels/momentumTransportModels/lnInclude \
    -I$(LIB_SRC)/MomentumTransportModels/compressible/lnInclude \
    -I$(LIB_SRC)/MomentumTransportModels/phaseCompressible/lnInclude \
    -I$(LIB_SRC)/finiteVolume/lnInclude \
    -I$(LIB_SRC)/meshTools/lnInclude

LIB_LIBS = \
    -lfluidThermophysicalModels \
    -lreactionThermophysicalModels \
    -lspecie \
    -lODE \
    -lfiniteVolume \
    -lphaseSystem \
    -leulerianInterfacialModels \
    -lmeshTools
```



NTNU

Norwegian University of
Science and Technology