



Advanced Modelling of Dense Polydisperse Fluidized Powders via Hybrid CFD and Population Balance Modelling

Akpan Michael Effiong
Department of Chemical Engineering
University College London

A thesis submitted for the degree of
Master of Philosophy
of
University College London

January 22, 2019

I, Akpan Michael Effiong confirm that the work presented in this thesis is my own. Where information has been derived from other sources, I confirm that this has been indicated in the thesis.

Abstract

Fluidization is used in several industrial processes, for example in waste disposal, food processing, pharmaceutical applications, energy conversions and so on. Although virtually every industrial plant contains units which treat multiphase polydisperse systems (MPS), designing them is still subject to great uncertainties; this is because MPS involve many physical and chemical phenomena that occur simultaneously and are hard to model: particles move in space and may shrink, grow, aggregate or break (in some cases new particles may also form). The unit behaviour and the product quality strongly depend on these competing phenomena, on the system fluid dynamics and on the unit geometry and size. This research project focuses on fluidized beds (FB) and aims to implement, test and validate an advanced computational fluid dynamics (CFD) model able to describe the behaviour of dense polydisperse fluidized powders and the evolution of their particle size distribution (PSD) in space by coupling CFD with population balance modelling (PBM) and by accounting for size-changing phenomena such as aggregation and breakage. To begin our work, we derived useful mathematical models including the generalized population balance equation, DQMOM and QMOM multifluid models and size-based aggregation and breakage kernels based on the kinetic theory of granular flow. In this work, we have implemented, in the commercial code Fluent, the QMOM Model, employing quadrature formula based on two, three and four nodes. Our model included convection in physical space, particle aggregation and breakage. We tested that the model had been correctly implemented by comparing model results with those reported in Fan et al. (2004). In terms of trend, results from our simulations showed good agreement with those reported by Fan et al. (2004). However, we noticed some quantitative differences in the results. These differences are due to the different constitutive equations used in CFD codes Fluent and MFIX, employed in this work and Fan et al. (2004) respectively. Furthermore, for simulations using four-node quadrature approximation, we were unable to replicate the trend as a result of the corruption of transported moment set. We observed similar corruption of the moments when we used higher order discretization schemes.

We tackled the problem of moment corruption by implementing moment correction algorithms in Fluent. Nevertheless, we observed that the moment correction algorithm was effective in the QMOM model with the moment set transported with the same velocity compared with QMOM model with different velocities for the moment sets. We observed that in the latter the corrupt moment set is used to compute 'corrupt' velocity fields for the quadrature classes which in turn complicates the solution. In order to further test the robustness of the implemented model we ran segregation tests using the QMOM model and compared them with experimental results. Our numerical results for the quadrature nodes and volume fractions as well as axial segregation profile was in good agreement with experimental results.

Impact Statement

Particle technology is a recurrent theme in the chemical engineering industry especially within the pharmaceutical space where particle size is key to defining the product quality and properties. The research documented in the pages that follow serves fundamentally as a bedrock for further researches in the aspect of particle sizing. In the pharmaceutical industry, a lot of long, arduous processes are involved in moving basic concepts on to the dispensary shelves of the pharmacists. Among these processes, particle sizing is arguably one of the salient steps to be concerned about. We know that many industrial units use multiphase polydisperse systems and designing them is still subject to great uncertainties. Predicting the size of particles or drug forms pose a big challenge to the process industry. This research helps to provide an alternative to expensive and time demanding pilot designs for technology innovations. For example the variation in an operating condition and how it affects the particle size distribution of the multiphase system should, in the future, be easily ascertained by a working model which has been tested to be reliable and robust. This research also impacts on the need to further develop a predictive model for particle sizing - one which can encompass the effects of size-changing phenomena such as aggregation, breakage, growth, nucleation and so on, thereby, presenting a relatively cheaper, faster and more reliable way to predict the impact of process conditions to particle size distribution, the process of initiating new technologies and also making existent processes better.

Contents

1	General Introduction	1
1.1	The Concept of Fluidization Technology	1
1.2	Computational modelling of fluidized systems	4
1.3	Motivation and Research Objectives	6
1.4	Thesis Outline	7
2	Review of Related Literature and Mathematical Modelling of Dense Polydisperse Fluidized Powders	8
2.1	Literature Review: Introduction	8
2.1.1	Related Literature	9
2.2	Modelling of Disperse Multiphase Flows: Introduction	12
2.3	Basics of Polydisperse Multiphase Flows Modelling	13
2.3.1	Definitions	13
2.3.2	Number Density Function	14
2.3.3	Number Density and Moments of the NDF	14
2.3.4	Size-based and Volume-based NDF	15
2.3.5	Convection Velocities in Particle State Space	16
2.4	The Generalized Population Balance Equation	16
2.4.1	Solving the Generalized Population Balance Equation	18
2.4.2	Method of Moments	19
2.4.3	The Population Balance Equation (PBE) as moment transform of the GPBE	21
2.5	Closures based on Quadrature Approximations	23
2.5.1	The Direct Quadrature Method of Moments (DQMOM)	24
2.5.2	Quadrature Method of Moments (QMOM)	31
3	Mathematical Modelling of Aggregation and Breakage	34
3.1	Introduction	34
3.2	Breakage and Aggregation Equations	35
3.2.1	The Breakage Equation	35
3.2.2	The Aggregation Equation	36
3.2.3	The Population Balance Equation with Aggregation-Breakage Equation	38
3.2.4	Moment Equation of the Generation term h_n	40
3.3	Modelling the Aggregation and Breakage Kernel	43
3.3.1	The Aggregation Kernel	44

3.3.2	The Breakage Kernel	45
3.3.3	The Daughter Distribution Function	46
4	Model Description and Methodology	49
4.1	Mathematical Models	49
4.1.1	The Multi-Fluid Model for Gas-Solid Flow	49
4.1.2	DQMOM Multi-Fluid Model	53
4.1.3	Aggregation and Breakage Models	55
5	Results and Discussions	57
5.1	Model Implementation	57
5.2	Implementation Check	58
5.2.1	Constant Kernel	58
5.2.2	Kinetic Kernel	68
5.3	Correcting Corrupted Moments	71
5.3.1	Implementation of Moment Correction Algorithm	73
5.3.2	Results from Moment Correction	74
5.4	Segregation of Fluidized Powders	78
5.4.1	Boundary and Initial Conditions	79
5.4.2	Fluidized Bed Segregation Results	79
6	Conclusions and Future Work	84
6.1	Future Work	86
	Appendices	88

List of Tables

3.1	Constitutive relations for the solids collision parameters	45
5.1	Initial values of particle diameter(s_i) and solid-phase volume fractions (ϕ_{si})	59
5.2	PSD at the middle of the bed at time 10s for Case 2 using N=3	65
5.3	PSD at the middle of the bed at time 10s for Case 2 using N=2	65
5.4	PSD at the middle of the bed at time 10s for Case 3 using N=3	68
5.5	PSD at the middle of the bed at time 10s for Case 3 using N=2	68
5.6	Initial values of the quadrature nodes and weights obtained experimen- tally at $t = 0s$ with voidage of 0.4 (Mazzei et al., 2009)	79

List of Figures

1.1	Primary single forces (Gibilaro, 2001)	2
1.2	Fluidization regimes (Kunii and Levenspiel, 2013)	3
1.3	Some industrial applications of Fluidized beds (Lettieri, 2014)	3
1.4	Some industrial applications of Fluidized beds (Geldart, 1973)	4
1.5	Fluidization behaviour of the Geldart fluid-particle groups (Kunii and Levenspiel, 2013)	5
3.1	The additive nature of volume-based vs. size-based intrinsic particle property	47
5.1	[A] Nodes and [B] Volume fractions for Case 1 at time 0s for N = 2 . . .	60
5.2	[A] Nodes and [B] Volume fractions for Case 1 at time 10s for N = 2 . . .	60
5.3	volume-average mean particle size vs. time for Case 1 and N = 2 and 3 . .	61
5.4	difference- and PASE- checks for three-node quadrature approximation using second-order upwind discretizaion scheme at 0.05s - Case 1	63
5.5	volume-average mean particle size vs time for Case 2, second order upwind scheme (2 Nodes)	64
5.6	volume-average mean particle size vs time for Case 2, first order upwind scheme (3 Nodes)	64
5.7	volume-average normalized moments for case 2 for two-node quadrature formula	65
5.8	volume-average mean particle size vs time for Case 3, second order upwind scheme (2 Nodes)	66
5.9	volume-average mean particle size vs time for Case 3, first order upwind scheme (3 Nodes)	67
5.10	volume-average mean particle size vs time for Case 3, second order upwind scheme (3 Nodes)	67
5.11	volume-average mean particle size vs time for Case 4, 1st and 2nd order upwind scheme (2 Nodes)	68
5.12	volume-average mean particle size vs time for Case 4, first order upwind scheme (3 Nodes)	69
5.13	contour of granular temperature for (a)solid 1 (b) solid 2 at 10s for Case 4 (2 Nodes)	70
5.14	volume-average mean particle size vs time for Case 5	70
5.15	volume-average mean particle size vs time for Case 5 using N=4	71
5.16	Flowchart to show how the invalid moments are identified and corrected .	73
5.17	Graph to show how d_{32} varies with time for Case 1 with moment correction algorithm	74

5.18	Graph to show how d_{32} varies with time for Case 1 without moment correction algorithm	75
5.19	Contour of volume fraction of Node 1 for four-node quadrature Case 2 at 0.50s and 0.91 s. (White contours show the presence of moment corruption in the affected cells)	76
5.20	Graph to show how d_{32} varies with time after implementing moment correction for Case 2 simulations (particles transported at same velocity) . .	76
5.21	Graph to show how d_{32} varies with time after implementing moment correction for Case 3 simulations (particles transported at same velocity) . .	77
5.22	Diagrammatic representation of the segregation experiment	79
5.23	Quadrature Nodes for experimental and simulation results at pseudo-stationary conditions (t = 0s) for fluidization velocity = 0.10m/s	80
5.24	Axial Segregation Profile for Quadrature Node 3 at Pseudo-stationary conditions (t = 10s) for fluidization velocity = 0.10 m/s	81
5.25	Contour of Volume fraction for Quadrature Node 3 at pseudo-stationary conditions (t = 10s) for fluidization velocity = 0.10 m/s	82
5.26	Axial Segregation Profile for Quadrature Node 1 at Pseudo-stationary conditions (t = 10s) for fluidization velocity = 0.10 m/s	83

Chapter 1

General Introduction

In this chapter we briefly introduce the concept of fluidization, take a look at the computational modelling of fluidized suspensions, motivate the research work and summarise the aims of our study. We end this chapter with a brief outline of the thesis.

1.1 The Concept of Fluidization Technology

Fluidization is defined as “the operation by which solid particles are transformed into a fluidlike state through suspension in a gas or liquid.” In a fluidized bed, stationary solid particles with sizes within the range $15\mu\text{m} < s < 6\text{mm}$ are placed in a vertical vessel over a porous base (*distributor plate*) through which fluid is passed upward. This is referred to as the *fixed bed*. The fluid is passed upward at an increasing velocity which makes the particles vibrate and move apart in restricted regions. This is the *expanded bed*. At a point the drag force f_d , which refers to the frictional force between the particles and the fluid, equals the effective weight of the particle W_e (this equals the real weight f_g minus the buoyancy force f_b). This point is the *minimum fluidization point*. At this point, the solid particles no longer rest on the base and can be handled like fluids. We say the bed is fluidized. The *minimum fluidization velocity* U_{mf} is the minimum velocity of the fluid needed to fluidize the solid particles. (Kunii and Levenspiel, 2013). Figure 1.1 shows the forces acting on a single particle in a fluidized bed.

In a liquid-solid fluidized bed, increasing the fluid velocity beyond U_{mf} causes the bed to expand smoothly and remain statistically homogeneous. We say the bed is *homogeneous or uniform* and the corresponding fluidization regime is the *homogeneous or particulate* regime. This regime is observed in a gas-solid fluidized bed under special conditions and for very small and light particles. However, in most gas-solid fluidized beds, increasing the fluid velocity beyond U_{mf} causes *bubbles* which refers to regions within the bed that are void of particles as a result of the excess fluid. This type of system is referred to as *bubbling* and the corresponding fluidization regime as *bubbling or aggregative* regime. The fluidization velocity at which the first bubbles form is called the minimum bubbling velocity U_{mb} . The bubbles merge as they rise up the bed. If the bed is deep enough, the bubbles may eventually grow to fill up the cross-sectional area of the vessel. This is

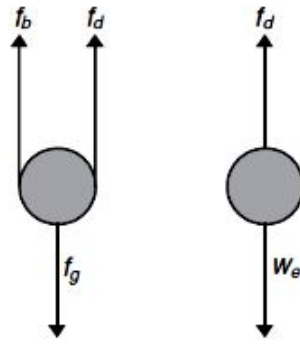


Figure 1.1: Primary single forces (Gibilaro, 2001)

called the *slugging* fluidization regime. Furthermore, increasing the fluid velocity beyond the terminal velocity will result in the particles moving faster and being more chaotic. Streamers and clusters of particles continuously form and break, and the overall structure of the bed becomes more homogeneous. This refers to the *turbulent* regime. Any further increase of the fluidization velocity beyond this point will lead to a lean fluidized bed with *pneumatic transport*. Figure 1.2 shows the different fluidization regimes we have just described.

Fluidization plays a very important role in chemical process industries and in common processes such as food processing, gasification, combustion, waste disposal and tablet coating in pharmaceutical industries. Their excellent features such as enhancing good contact surface areas between the solid and the fluid, efficient heat and mass transfer, and suitable mixing tendencies make them an attractive technology in process industries (Mazzei, 2011). For example, processes involving gas/gas reactions catalyzed by the solid (e.g. the catalytic cracking of oil). Some industrial applications of fluidized beds are summarised in Figure 1.3.

How fluid-particle systems behave is dependent on the physical properties of the materials and on the operating conditions. We have discussed how changes in the fluid velocity changes the fluidization regimes or behaviour of the particles. Let us also see how the physical properties of the fluidized powders can play a significant role in characterizing the behaviour of fluidized suspensions.

By carefully observing the behaviours of several fluidized systems, Geldart (1973) proposed a classification where fluidized solids are classified based on two coordinates, once the fluid and air temperature are assigned. The coordinates are the mean particle diameter of the powder and the difference between the particle and the fluid densities. Figure 1.4 shows an example for air at ambient temperature. We shall briefly consider the four different powder classification by Geldart.

Group C solids are cohesive and characterized by very small particles (between $20\mu\text{m}$ and $50\mu\text{m}$). ‘Normal’ fluidization of this class of powders is extremely difficult. This is

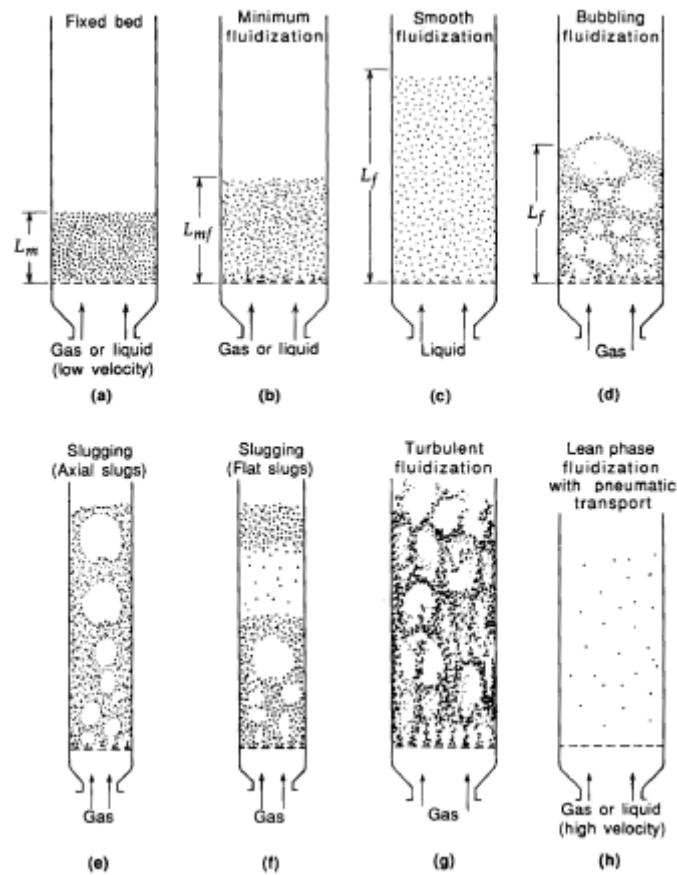


Figure 1.2: Fluidization regimes (Kunii and Levenspiel, 1973)

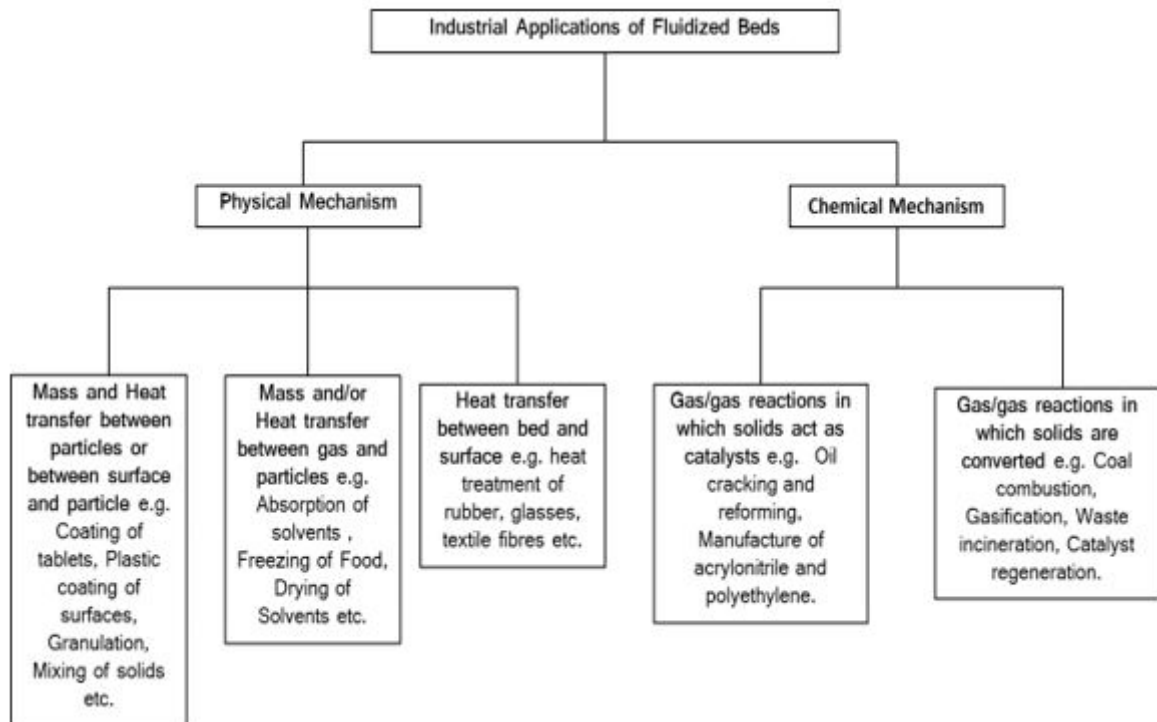


Figure 1.3: Some industrial applications of Fluidized beds (Lettieri, 2014)

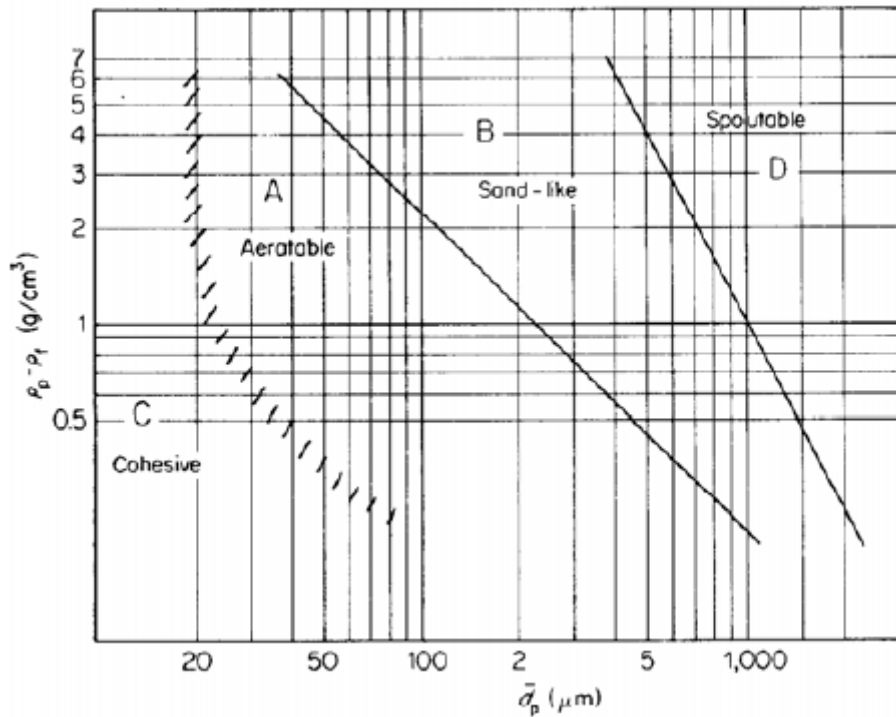


Figure 1.4: Some industrial applications of Fluidized beds (Geldart, 1973)

because the existing high inter-particle cohesive forces are greater than those which can be exerted by the fluid on the particles. Hence, the particles tend to stick together and do not fluidize properly. Since the fluid cannot separate the particles, fluid bypasses through channels extending from distributor to bed surface. The use of mechanical stirrers and vibrators help improve the fluidization quality by breaking the channels as they form.

Group A solids have a small mean size and/or low particle density (less than about 1.4 g/cm^3). They expand uniformly before the commencement of bubble formation. They are referred to as *aeratable*. The fluidization quality of this group of powders are affected by fluid dynamics and inter-particle forces. Some researchers report that the uniform structure found at low fluid velocities is largely due to the inter-particle forces (Massimilla et al., 1972; Mutsers and Rietema, 1977; Rietema and Piepers, 1990)

Groups B and D solids bubble as they fluidize. Thus, U_{mf} and U_{mb} coincides. These systems are purely fluid dynamic since the inter-particle forces are negligible. The difference between these two groups of powders is that whereas in Group B systems most bubbles rise through the bed more quickly than the interstitial fluid, in Group D systems the opposite occurs. Figure 1.5 shows the fluidization behaviour of the four groups.

1.2 Computational modelling of fluidized systems

The early development of fluidization technology was mostly characterized by the use of experimental correlations. This method can only help design standard units which the

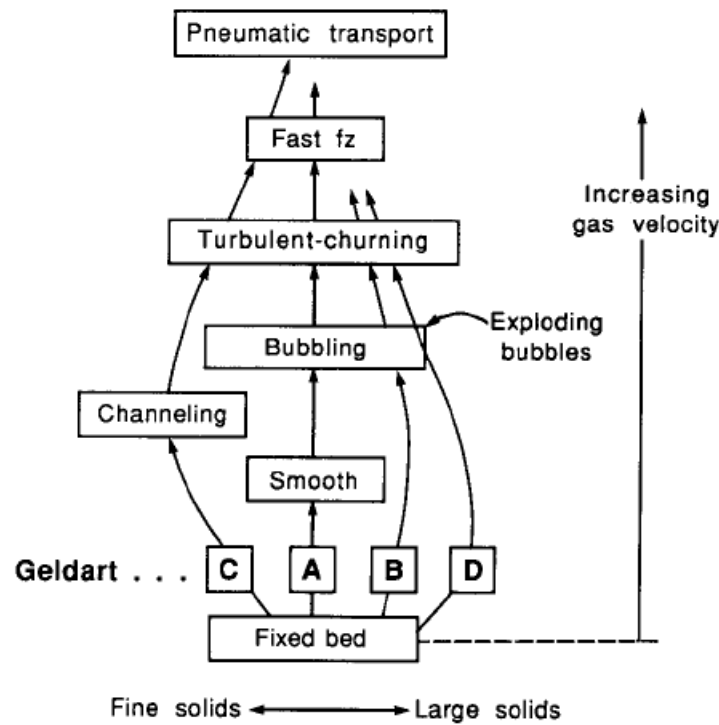


Figure 1.5: Fluidization behaviour of the Geldart fluid-particle groups (Kunii and Levenspiel, 2013)

experiments relate to. Moreover, one can unarguably conclude that optimizing a fluidized system via experiments and correlations is quite a time-consuming and expensive venture (Buffo and Marchisio, 2014). They lack the flexibility of telling us how changes not directly related to the experiment can affect fluid dynamics and performance. The use of experiments to show how certain variables alter the bed performance and behaviour might be very expensive. For example, how vessel geometry affects the fluidized bed performance will be financially demanding. Hence, the shift towards more predictive methods for the behaviour of dense fluidized suspensions.

Recently, there has been an increased understanding of the mechanisms governing the highly complex fluid flow phenomena in fluidized beds. This is as a result of increased interest in computational modelling of fluidized bed systems. In order to properly design and operate fluidized beds, accurate models providing detailed information on important bed phenomena such as the bed hydrodynamics is necessary (Di Renzo and Di Maio, 2007). Researchers have developed mathematical models that describe the fluid dynamics and, in general, the behaviour of multiphase flows (in this case, of fluidized beds). These models need to be solved, tested and validated for reliability.

Models that describe multiphase gas-solid flows are based on two major approaches: the *Eulerian-Lagrangian* and the *Eulerian-Eulerian* approaches. The former is a detailed and more reliable representation of the gas-solid system as it tracks individual particles within the system. However, the efficiency of this method is at the expense of high computational cost. For modelling a dense fluidized bed, for example, this method will be less preferable due to its costly computational demands. The Eulerian-Eulerian model on the other hand is less computationally expensive and treats both solid and fluid phases as

interpenetrating continua. This model provides the required information by solving averaged equations of motion (Mazzei, 2008).

The numerical solution of the Eulerian averaged transport equations is known as *computational fluid dynamics* (CFD). CFD has led to reductions in the cost of product and process development activities, reduced the need for physical experimentation and increased design reliability (Davidson, 2003). CFD has become an important tool in solving many complex problems of both academic and industrial interest. In fluidization engineering, a better understanding of fluid-solid interactions and macroscopic phenomena in particulate systems has been made possible via CFD.

1.3 Motivation and Research Objectives

Research efforts to investigate gas-solid fluidized powders have evolved over time: from studies of monodispersed systems (Sinclair and Jackson, 1989; Ding and Gidaspow, 1990; Kuipers et al., 1993; Hrenya and Sinclair, 1997; Pain et al., 2001; Lettieri et al., 2002; Cammarata et al., 2003; Gelderbloom et al., 2003; Lettieri et al., 2004; Mazzei and Lettieri, 2008) to bidisperse powders allowing for investigation of segregation and mixing in fluidized beds (Van Wachem et al., 2001; Wirsum et al., 2001; Howley and Glasser, 2002; Huilin et al., 2003; Gera et al., 2004; Cooper and Coronella, 2005; Qiaoqun et al., 2005; Huilin et al., 2007; Owoyemi et al., 2007; Mazzei et al., 2010). Most CFD calculations for gas-solid fluidized beds assume the solid phase to be monodisperse and in some cases bidisperse. However, in reality, most fluidized beds feature polydispersity and designing them is still subject to great uncertainties. This is because multiphase polydisperse systems, for example, in a fluidized bed, involve chemical and physical phenomena that occur simultaneously and are hard to model. The particles move in space and may shrink, grow, aggregate or break and in some cases new particles may also form. The unit behaviour and the product quality strongly depend on these competing phenomena, on the system dynamics and on the unit geometry and size.

Recently, researchers have tried to account for polydispersity and other particle-related phenomena such as aggregation and breakage by coupling the population balance equation (PBE) with a multi-fluid model. However, very few have considered modelling a dense polydisperse fluidized bed with focus on both particle and fluid dynamics. One work which fits this category is that of Fan et al. (2004) where a CFD model for polydisperse gas-solid fluidized bed was developed and implemented.

In this work, we will implement a state-of-the-art hybrid CFD-PBE model within the commercial CFD code, Fluent and verify its ability to predict correctly the evolution of the particle size distribution (PSD) of dense polydisperse fluidized powders. This study will focus on the Eulerian modelling approach and the PBE will be solved using quadrature based moment methods. In the process, we will:

1. Derive the Generalized Population Balance Equation from first principles.
2. Analyse the quadrature based method of moments in steps to guide the reader through the theoretical analysis of the methods. Only the Direct Quadrature Method

of Moments (DQMOM) and the Quadrature Method of Moments (QMOM) will be discussed.

3. Derive the Breakage and Aggregation Modelling equations from first principles.
4. Derive expressions based on the literature for aggregation and breakage kernels to be used in this work.
5. Implement Quadrature based method of moments in Fluent. (Note: The implementation of DQMOM and QMOM with same velocity for the quadrature nodes in Fluent is not part of the scope of this work as these are already implemented as default models in Fluent 17.2 which we will be using for our simulations in this work.
6. Implement Moment correction for Quadrature Method of Moments (for quadrature nodes transported with same and different velocities).

We will also look at the prediction of segregation using the QMOM model.

1.4 Thesis Outline

In Chapter 2, we take a look at principal mathematical theories underlying the modelling of multiphase polydisperse fluidized suspensions, the generalized population balance equation, QMOM and DQMOM. We also briefly discuss related literature.

In Chapter 3, we discuss the mathematical theories underlying the modelling of birth and death functions due to aggregation and breakage in gas-solid fluidized beds.

In Chapter 4, we present the methodology for our work. This includes a brief description of the mathematical models employed in this work.

In Chapters 5, we present and analyse simulation results obtained. All simulation results will be documented in this chapter under relevant sections.

In Chapter 6, we conclude our findings and present recommendations for future work.

Chapter 2

Review of Related Literature and Mathematical Modelling of Dense Polydisperse Fluidized Powders

This chapter is mainly in two parts: a section which briefly reviews related literature and another which discusses the principal mathematical theories underlying the modelling of multiphase polydisperse fluidized suspensions. The literature reviewed are those that have covered the scope of work relevant to our research. The scope of our work include:

- 1) Gas-solid fluidized bed systems or fluidized suspensions ;*
- 2) Hybrid CFD and Population Balance Models (PBM) ;*
- 3) Quadrature Based Moment Methods;*
- 4) Eulerian modelling approach.*

Works reviewed in this section are those that have considered all the elements listed above in a single project or as part of a series of progressive research work.

We present the population balance equation needed to describe the flow of polydisperse fluidized powders. We also discuss the method of moments for solving the PBE and the associated problem of closure. To solve the problem of closure, we introduce the quadrature method of moments.

2.1 Literature Review: Introduction

Predicting the dynamics and behaviour of fluidized beds is a continuing research area. The processes involved in a fluidized bed are quite complex making it difficult to predict and control fluidized suspensions. Intricate chemical and physical processes occur at the same time affecting the properties of the bed: they can shrink, break, aggregate or form new particles via nucleation. These phenomena affect the product quality to a large extent. Therefore, in order to design these units, the process engineer should take cognizance of

the effect of these competing phenomena while modelling.

Computational Fluid Dynamics which was introduced in Chapter 1 has played a very key role in comprehending the behaviour of fluidized bed suspensions and multiphase flows in general. In this regard, there has been a lot of improvement in terms of accuracy of fluid dynamic models over the past decade, thereby increasing the interest of industry in this technique. However, more still need to be done in terms of developing complex models that can accurately predict the complex systems involved in multiphase systems (in our case fluidized beds) such as testing, verification and validation of these models to make CFD utterly reliable as a design tool to mimic the behaviour of fluidized suspensions.

Researchers have made several contributions which has led to CFD models that can represent the behaviour of fluidized beds that feature polydispersity. As we already know, most industrial processes feature a PSD that changes in time and space in the light of the chemical and physical processes that characterize them. These changes in PSD are associated with phenomena like aggregation and breakage which is still an on-going area of research.

We shall briefly review some of the works that have focused on polydisperse fluidized suspensions accounting for size changing phenomena due to aggregation, breakage, growth, mixing etc. The use of hybrid CFD and population balance models (PBM) in these works will also be of interest.

2.1.1 Related Literature

Fan et al. (2004) were the first to implement the DQMOM in a CFD environment to study the evolution of PSD in a fluidized bed. They studied the size changing phenomenon as a result of aggregation and breakage in fluidized bed polymerization reactors using a CFD-PBM hybrid model. The evolution of the PSD was modelled using the PBE which was coupled to both the continuity and the momentum balance equations. The DQMOM, which shall be discussed in section 2.5.1 of our work, was implemented in MFIX, an open-source multifluid CFD code, to reproduce the effect of particle aggregation and breakage. Two different kernels were tested: the constant kernel and a kinetic kernel developed from the kinetic theory of granular flow (KTGF) and results were obtained for two-, three- and four- quadrature node approximations. For constant aggregation and breakage, the results were independent of the number of nodes. Whereas for the kinetic kernel, though the results for the volume-average mean diameter for two, three and four nodes were similar, some differences were observed in the defluidization dynamics. Computational time also increased with increasing number of nodes as expected. The study was mainly qualitative without subsequent validation with experiments. It was mainly based on proof of concept. The trends from the results obtained showed that the method adopted is effective in modelling the evolution of the PSD of dense fluidized suspensions due to aggregation and breakage. Three cases were simulated for the constant kernel: Case 1 - No aggregation and breakage, Case 2 - Aggregation dominant and Case 3 - Breakage dominant. The volume-averaged mean particle diameter (d_{32}) was tracked over time for each of these cases and as expected for Case 1 d_{32} remained fairly constant, for Case 2 d_{32} showed

a gradual increase with time which led to the defluidization of large regions of the bed and for Case 3, the bed became more dilute as d_{32} became more and more smaller due to excessive breakage. For the kinetic kernel, two cases were simulated: Case 4 - Aggregation dominant and Case 5 - Breakage dominant. These settings were achieved by manipulating the values of the breakage and aggregation success factors. Case 4 showed an increase in d_{32} with time leading to defluidization in the bottom part of the bed. Unlike the constant kernel, where aggregation took place even after most part of the bed became defluidized, here, aggregation and breakage stopped as the granular temperature at the point of defluidization was zero. For Case 5 a similar trend as observed in the constant-kernel-breakage-dominant case was reported.

Mazzei et al. (2009) implemented the DQMOM within Fluent, a commercial CFD code, to study the mixing of two inert polydisperse fluidized powders initially segregated. A two-node quadrature approximation was employed where each quadrature node represented a unique secondary phase advected with its own velocity. A simple case of perfect mixing in the absence of aggregation and breakage was studied. Hence, the evolution of the PSD in the study was mainly due to mixing. The investigation comprised both experiments and computational modelling. Initial results obtained from the DQMOM transport equation correctly predicted the weights whereas the nodes were not. In order to correct the observed result, a diffusion term was introduced to yield source terms in the transport equations that ensured that the nodes were correctly predicted.

Mazzei et al. (2012) implemented the DQMOM model based on volume density function, rather than on number density function, therefore, making it possible to deal with volume fractions directly instead of number densities. A simple case was tested where the powders were inert and initially segregated. The particles do not break or aggregate; change in PSD was mainly due to mixing. They considered two-, three-, and four-quadrature nodes to compare their performances and predictions with experimental data. The two-node quadrature approximation gave good results with only a 1% maximum deviation in predicting the weights and nodes. The three-node quadrature predicted the nodes with a deviation of 6% but predicted the weights poorly with a 60% maximum deviation. Corruption of higher order moments occurred using the four-node quadrature, thereby crashing the simulation. This observation worsened with the use of higher order spatial discretization schemes. The problem was associated with the convective terms in the moment transport equations. An attempt to correct the problem by implementing a corrective algorithm by McGraw (2006) was unsuccessful. They suggested finding the moment flux using only nodes and weights and also using higher-order schemes only for the weights.

Yan et al. (2012) incorporated the polymeric multilayer model (PMLM), a single particle growth model, into the CFD-PBM model to simulate olefin polymerization. The PMLM was introduced to study intraparticle mass and heat transfers which are fundamental in modelling and can also influence the outcome of simulation results. The integrated CFD-PBM-PMLM model was solved to predict the evolution of the PSD. Five cases were studied including pure particle breakage and pure particle aggregation. The bed expansion heights from the simulations were compared to those from experiments and were

found to be in good agreement. It was concluded that although the model could be used to simulate flow fields in FB polymerization reactors, the intraparticle mass transfer limitation affected the reactor flow fields. Conversely, fluidization results stayed almost the same for different intraparticle heat transfer limitations.

Li et al. (2013) studied the behaviour of a turbulent gas-solid flow and the reaction kinetics in a polydisperse fluid catalytic cracking (FCC) riser reactor. A CFD-PBM model was used to simulate the hydrodynamics of the solid catalyst particles and the 14-lump reaction kinetics model was used to describe the complex FCC reactions that take place in the unit. The gas-solid flow was modelled using the Eulerian-Eulerian two-fluid model. They considered two cases: Case1- featuring single component of catalyst particles with continuous PSD without aggregation and breakage and Case2- PSD of catalyst particles undergoing aggregation and breakage. For Case1, d_{32} remained constant with the FCC process whereas for Case2, a breakage dominant trend was observed as d_{32} decreased with time. The PSD behaviours for both cases were also observed. Similar PSD curves at different times were reported for Case1 whereas for Case2 the PSD curves flatten out with time. They also observed that particle diameter affected the interphase exchange coefficient. Therefore, in Case2, due to aggregation and breakage the particles experience different drag forces leading to complex particle collisions and the promotion of particle backmixing. This made it easy to form particle clusters which led to reduced particle velocity. In conclusion, they reported that aggregation and breakage results in more complex flow fields in the riser.

Akbari et al. (2015) coupled a 2D CFD-PBM/DQMOM model to simulate particle growth in an industrial gas phase fluidized bed polymerization reactor. The effect of aggregation and breakage were neglected in this study. Simulation results showed a change in PSD due mainly to particle growth. Larger particles were observed to settle at the bottom of the bed while smaller particles migrated to the top. In order to validate the model accuracy and capability, grid independence study and time step analysis were carried out and results were compared with industrial data sourced from an industrial linear low density polyethylene (LLDPE) gas-phase polymerization reactor. The minimum fluidization velocity was also assessed for the polydisperse bed in order to observe the effect of the PSD on this important hydrodynamic bed design parameter. A graph of pressure drop in the bed against superficial gas velocities revealed three regions of fluidization. The fixed bed region where the fluidization velocity is less than the U_{mf} of the small particles. The transient fluidization region where segregation of small particles is seen to occur. This transient fluidization results gradually in mixing as the fluidization velocity is further increased. Lastly, a vigorous-bubbling region was also established where the fluidization velocity was beyond the U_{mf} of the bigger particles leading to a well-mixed bed. The results reported for the bed were in good agreement with the fluidization structure proposed by Olivieri et al. (2004)

Che et al. (2015) studied the flow behaviours and PSD of polyethylene in a pilot-plant fluidized bed reactor (FBR) using a CFD-PBM/QMOM modelling approach. The quadrature method of moments (QMOM) unlike the DQMOM assumes that the secondary phases are all advected with the same velocity (reader is referred to section 3.4.2). The polymer PSDs

were tracked using an Euler-Euler two-fluid model with polymerization kinetics coupled with population balance. Particle growth, aggregation and breakage were accounted for in the study. The predicted simulation results for the pressure drop and temperature were compared with experimental data and found to be in good agreement. These results were used to verify the model as being feasible for describing the flow behaviours of ethylene polymerization reaction in a fluidized bed reactor. Six cases were tested to assess the CFD-PBM model, some involving breakage, others involving aggregation. But, none of the cases considered simultaneous polymer aggregation and breakage. The effects of ethylene polymerization, the particle kinetics and the non-pelletizing polyethylene process (NPPP) on the flow behaviour and the PSD were also reported. Although these results were not validated with experiments, they helped to understand the flow behaviours and the evolution of PSDs in industrial scale FBRs. In the case involving breakage, significant bed expansion was observed compared to the cases involving growth and aggregation of the polyethylene particles. This obvious change was due to the decrease in polymer particle size which caused the particles to be easily lifted by the fluidizing gas to a higher height leading to particle entrainment.

Yao et al. (2015) studied the effect of three different methods of moments for solving the PBEs in the CFD-PBM model for polydisperse polymerization FBRs characterized by simultaneous growth and aggregation. The three representative methods used in the study are: QMOM, DQMOM and a new method recently proposed by Gu et al. (2009) namely the fixed pivot quadrature method of moments (FPQMOM) in which the characteristic abscissas are specified at the beginning of the simulation and also kept constant throughout the simulation. In the FPQMOM, an approximation different from that used in QMOM and DQMOM was used to represent the PSD and based on the approximation, $2N$ set of moments can be written in matrix form to form the Vandermonde equation set. A special algorithm different from the product difference (PD) algorithm employed in QMOM and DQMOM is used to solve the resulting Vandermonde equation set to obtain the weights. The FPQMOM predicts a faster rate of particle enlargement for bigger particle size clusters, whereas the QMOM and DQMOM gives the opposite trend. The three MOMs predicted a reasonable pressure drop variation, time-averaged flow field and temperature distribution. Also, the computational time for the FPQMOM was also compared with those of DQMOM and QMOM and found to be shortest.

2.2 Modelling of Disperse Multiphase Flows: Introduction

Real systems in fluidized bed reactors have a PSD which play a significant role in the performance of fluidized beds. The particles may aggregate or break and this will affect the product quality and unit behaviour of the system. Therefore such system or particle behaviour needs to be factored in during design and subsequent scale-up procedures. The prediction of the evolution of the PSD is cardinal in many engineering applications but the simulation and scale-up of such processes is very challenging because of the intricate interactions between mixing at different scales, growth, aggregation and breakage (Marchisio et al., 2003b). The evolution of the particle size distribution is described by

solving customary continuity and linear momentum balance equations alongside a population balance equation (PBE).

A large number of the equipment used in the chemical process industries make use of multiphase flow systems. For example, fluidized beds, pipelines in the oil and gas industries, bubble columns and so on, but to mention a few. These systems are also designed, optimized and scaled up using effective computational tools to simulate their operations. The modelling of polydisperse multiphase systems is in two parts:

- i) accounting for the disperse phases: This includes aggregation, breakage, and collision of particles. Here, the PBE is used to describe the evolution of the discrete particles in time and space. The PBE is often written in terms of a density function which provides information on the population of the system distributed over a particular property of interest in a given volume at time t ;
- ii) accounting for the multiphase fluid dynamics: This focuses on the interactions between the disperse (or secondary) phase(s) and the continuous (or primary) phase. Particles can interact between themselves as well as with the continuous phase. CFD models are used to describe the flow dynamics between the disperse and continuous phases. (Marchisio and Fox, 2013).

A bridge between these two approaches will help solve a large variety of polydisperse multiphase flow problems.

2.3 Basics of Polydisperse Multiphase Flows Modelling

Although the theory explained subsequently has applications in a much wider context (Chapman and Cowling, 1970; Randolph and Larson, 1971; Ramkrishna, 2000), we shall limit our discussion to particulate systems. We shall begin with introducing suitable density functions.

2.3.1 Definitions

Let us consider a polydisperse system of solid particles placed in a continuous medium in which each entity in the disperse phase can be identified by a certain set of properties referred to as *coordinates*. In this case, similar elements will have identical coordinates. Otherwise, the entities are distinguishable. These coordinates are classed as:

- i) *External coordinates*, $(\mathbf{x})$: This refers to the spatial coordinates; the three-dimensional coordinates (x, y, z) in physical space. Here, $\mathbf{x}$ is the position vector.
- ii) *Internal coordinates*, $(\boldsymbol{\xi})$: This refers to other properties of the particle other than physical location which completely describe the particle state. For example, size, age, surface area, volume etc. The symbol $\boldsymbol{\xi}$ represents the internal state vector.

If we consider $\Omega_{\mathbf{x}}$ and $\Omega_{\boldsymbol{\xi}}$ to be the domains of the position vector, $\mathbf{x}$ and the internal state vector, $\boldsymbol{\xi}$ respectively, we can define a particle state space (or domain), $\Omega_{\boldsymbol{\psi}}$ consisting of both the physical space, $\Omega_{\mathbf{x}}$ and the internal state space, $\Omega_{\boldsymbol{\xi}}$ where $\boldsymbol{\psi}$ is the particle state vector.

2.3.2 Number Density Function

From an Eulerian point of view, the number density function (NDF) is generally used for describing the disperse phase. It gives information about the properties of interest of a population of particles and their distribution within a given control volume. For example, if we consider an infinitesimal control volume in physical space about $\mathbf{x}(x, y, z)$ with physical space volume, $d\mathbf{x} \equiv dx dy dz$ and N number of internal coordinates, with internal state vector, $\boldsymbol{\xi} \equiv (\xi_1, \xi_2, \xi_3, \dots, \xi_N)$, the NDF, $\tilde{f}_n(\mathbf{x}, \boldsymbol{\xi}, t)$ will give the number of particles within the differential particle state space volume, $d\mathbf{x}d\boldsymbol{\xi}$. Therefore,

$$\tilde{f}_n(\mathbf{x}, \boldsymbol{\xi}, t) d\mathbf{x} d\boldsymbol{\xi} \quad (2.1)$$

gives the number of particles present within $d\mathbf{x}$ at the time t of interest having an internal state vector in the range $d\boldsymbol{\xi}$ around $\boldsymbol{\xi}$. The number of particles having an internal coordinate of exactly $\boldsymbol{\xi}$ is zero as the value of $\boldsymbol{\xi}$ is continuous within the internal state space volume. The NDF is a function of time, internal coordinates and external coordinates. The NDF is an average quantity. That is, it is smooth and can be differentiated with respect to spatial coordinates, internal coordinates and time.

The differential volume $d\psi$ in particle state space corresponds to $d\mathbf{x}d\boldsymbol{\xi}$. Therefore, we can rewrite (2.1) as:

$$\tilde{f}_n(\psi, t) d\psi \quad (2.2)$$

This yields the expected number of particles present within the volume $d\psi$ around ψ or the number of particles with particle state vector within the range $d\psi$ around ψ , at the time t of interest. We can say that the NDF is the expected number of particles per unit particle state space volume at the time t of interest.

In addition, we can also define the volume and mass density functions for the particle population of the disperse phase. If $v_p(\boldsymbol{\xi})$ represents the volume of the particle of internal state $\boldsymbol{\xi}$, then we can define the volume density function (VDF) as

$$\tilde{f}_v(\mathbf{x}, \boldsymbol{\xi}, t) = \tilde{f}_n(\mathbf{x}, \boldsymbol{\xi}, t) v_p(\boldsymbol{\xi}) \quad (2.3)$$

Similarly, the mass density function (MDF) can be defined as

$$\tilde{f}_m(\mathbf{x}, \boldsymbol{\xi}, t) = \tilde{f}_n(\mathbf{x}, \boldsymbol{\xi}, t) m_p(\boldsymbol{\xi}) \quad (2.4)$$

where $m_p(\boldsymbol{\xi})$ is the mass of the particle of internal state $\boldsymbol{\xi}$.

2.3.3 Number Density and Moments of the NDF

It immediately follows from the preceding section that the quantity $\tilde{f}_n(\mathbf{x}, \boldsymbol{\xi}, t) d\boldsymbol{\xi}$ is the number density, which represents the number of particles per unit physical volume with internal state vector in the range $d\boldsymbol{\xi}$ around $\boldsymbol{\xi}$. If the number density is integrated over all possible values of the internal state vector $\boldsymbol{\xi}$ within the internal state domain $\Omega_{\boldsymbol{\xi}}$,

$$n(\mathbf{x}, t) = \int_{\Omega_{\boldsymbol{\xi}}} \tilde{f}_n(\mathbf{x}, \boldsymbol{\xi}, t) d\boldsymbol{\xi} \quad (2.5)$$

we obtain the total number of particles per unit physical volume at a particular time t of interest. Equation (2.5) coincides with the zeroth-order moment, $\mathcal{M}_{\xi,0}(\mathbf{x},t)$ of the density function.

$$\mathcal{M}_{\xi,0}(\mathbf{x},t) = \int_{\Omega_{\xi}} \tilde{f}_n(\mathbf{x},\xi,t) d\xi \quad (2.6)$$

Similarly, we can define other arbitrary moments $\mathcal{M}_{\xi,k}(\mathbf{x},t)$ of the NDF by

$$\mathcal{M}_{\xi,k}(\mathbf{x},t) = \int_{\Omega_{\xi}} \xi_1^{k_1} \dots \xi_N^{k_N} \tilde{f}_n(\mathbf{x},\xi,t) d\xi \quad (2.7)$$

where $\mathbf{k} = (k_1, \dots, k_N)$ is a vector containing the order of the moments with respect to each of the components of ξ .

When we multiply equation (2.5) by $v_p(\xi)$ and $m_p(\xi)$, we obtain the volume fraction, ϕ and mass density, ρ_d respectively.

$$\phi(\mathbf{x},t) = \int_{\Omega_{\xi}} \tilde{f}_v(\mathbf{x},\xi,t) d\xi \quad ; \quad \rho_d(\mathbf{x},t) = \int_{\Omega_{\xi}} \tilde{f}_m(\mathbf{x},\xi,t) d\xi \quad (2.8)$$

2.3.4 Size-based and Volume-based NDF

If we consider a population of particles that is described by only one internal coordinate, in this case, size s , then we can define a size-based NDF $f_n(\mathbf{x},s,t)$ where the only component of the internal state vector is the particle size. Here, $f_n(\mathbf{x},s,t) ds$ represents the expected number density of particles with size between s and $s + ds$ at a particular time of interest. As reported in Section 2.3.3, the total number of particles per unit physical volume at a particular time t of interest can be written as:

$$n(\mathbf{x},t) = \mathcal{M}_{s,0}(\mathbf{x},t) = \int_{\Omega_s} f_n(\mathbf{x},s,t) ds \quad (2.9)$$

and the mean particle size is defined as

$$\bar{s}(\mathbf{x},t) = \frac{1}{n(\mathbf{x},t)} \int_0^{\infty} s f_n(\mathbf{x},s,t) ds = \frac{\mathcal{M}_{s,1}}{\mathcal{M}_{s,0}} \quad (2.10)$$

This average property of the distribution is defined with respect to the number of particles within the local volume $d\mathbf{x}$. Also, other particle size averages can be defined as the ratio $\mathcal{M}_{s,k+1}/\mathcal{M}_{s,k}$ for any value of k where

$$\mathcal{M}_{s,k}(\mathbf{x},t) = \int_0^{\infty} s^k f_n(\mathbf{x},s,t) ds \quad (2.11)$$

is the k -th order integer moment of the size-based NDF. For example, by setting $k = 2$ we obtain $s_{32} = \mathcal{M}_{s,3}/\mathcal{M}_{s,2}$ the sauter (or volume-averaged) mean particle size.

Another important internal coordinate worth considering for particulate systems is the particle volume v_p . If $\hat{f}_n(\mathbf{x},v_p,t)$ is the volume-based NDF, then we can define $\hat{f}_n(\mathbf{x},v_p,t) dv_p$ as the expected number of particles per unit physical volume with volume between v_p and

$v_p + dv_p$ at a particular time t of interest. If we consider the case where volume scales with the third power of the size such that $v_p = k_v s^3$ and $dv_p = 3k_v s^2 ds$, where k_v is the volumetric shape factor, then we can relate the size-based and volume-based NDF as follows:

$$\hat{f}_n(\mathbf{x}, v_p, t) dv_p = \hat{f}_n[\mathbf{x}, v_p(s), t] 3k_v s^2 ds = f_n(\mathbf{x}, s, t) ds \quad (2.12)$$

so that:

$$f_n(\mathbf{x}, s, t) = \hat{f}_n[\mathbf{x}, v_p(s), t] 3k_v s^2 \quad (2.13)$$

The ability to move from a volume-based NDF to a size-based NDF at will is useful because most chemical engineering designs are expressed in terms of system volume.

2.3.5 Convection Velocities in Particle State Space

If we consider the internal state space as being an abstract property space, we can define vector fields to describe the variation of the particle state vectors $(\mathbf{x}, \boldsymbol{\xi})$ with time. Here, we are concerned about the changes in the relevant coordinate spaces taking place gradually and continuously. For example, let us consider the size of particles increasing over time along the size coordinate. Particle size increment or decrease might be viewed as motion through an abstract size space similar to the particle motion in physical space. We have referred to this motion as convection making it convenient for us to define “velocities” $\mathbf{v}(\mathbf{x}, \boldsymbol{\xi}, t)$ and $\dot{\boldsymbol{\xi}}(\mathbf{x}, \boldsymbol{\xi}, t)$ for particles convected in physical and internal state space respectively. Similarly, the particle acceleration in this case, is change in particle linear velocity in the velocity state space and is denoted by $\dot{\mathbf{v}}(\mathbf{x}, \boldsymbol{\xi}, t)$.

2.4 The Generalized Population Balance Equation

The generalized population balance equation (GPBE) is a continuity statement written in terms of the number, volume or mass density functions. The GPBE basically accounts for the various ways in which particles can either appear in or disappear from a system. It is therefore an indispensable framework for studying dispersed phase system. It can be derived by writing a balance equation on the particles contained in some fixed subregion of the particle state space (Ramkrishna, 2000).

Here, we shall consider a generalized internal state vector $\boldsymbol{\xi}$ characterised by size space and velocity state space. Thus making the GPBE under consideration a quadrivariate (having four internal coordinates because velocity is a 3-dimensional vector). Let us consider a finite control volume Λ_ψ in the particle state space such that $\Lambda_\psi \subseteq \Omega_\psi$. Similarly, we can define

$$\Lambda_x \subseteq \Omega_x \quad ; \quad \Lambda_s \subseteq \Omega_s \quad ; \quad \Lambda_v \subseteq \Omega_v \quad (2.14)$$

in the physical space, size space and velocity state space respectively. Let each control volume be bounded by surfaces $\partial\Lambda_\psi$, $\partial\Lambda_x$, $\partial\Lambda_s$ and $\partial\Lambda_v$ respectively. The particle number balance taken about the finite control volume Λ_ψ per unit time can be expressed from first principles:

$$Acc = In - Out + Gen$$

where Acc is the accumulation, $In - Out$ is the net convective input and Gen is the generation. These terms are expressed as rates. The number of particles present $N(t)$ at time t in the control volume Λ_ψ is:

$$N(t) = \int_{\Lambda_\psi} \tilde{f}_n(\psi, t) d\psi \quad (2.15)$$

The accumulation term represents the rate of change of the number of particles contained in the control volume Λ_ψ . This is given by:

$$Acc = \frac{dN}{dt} = \frac{d}{dt} \int_{\Lambda_\psi} \tilde{f}_n d\psi = \int_{\Lambda_\psi} \partial_t \tilde{f}_n d\psi \quad (2.16)$$

where Liebnitz theorem is applied to the control volume Λ_ψ , which is assumed to be time independent. Let us account for convection via the boundary of each control volume. These will add up to give the net convective flow rate of particles entering Λ_ψ . For the control volume Λ_x , the net convective flow rate of particles entering through the boundary $\partial\Lambda_x$ is given as:

$$(In - Out)_x = - \int_{\Lambda_s} \int_{\Lambda_v} \int_{\partial\Lambda_x} \tilde{f}_n \mathbf{v} \cdot d\mathbf{S}_x ds dv \quad (2.17)$$

where $d\mathbf{S}_x$ is the differential surface vector normal to $\partial\Lambda_\psi$ and pointing outwards. Applying the Gauss theorem to equation (2.17) to transform the surface integral into a volume integral, we have:

$$(In - Out)_x = - \int_{\Lambda_\psi} \nabla_x \cdot (\tilde{f}_n \mathbf{v}) d\psi \quad (2.18)$$

where $\nabla_x \cdot$ is the divergence operator in the three-dimensional physical space. Although equation (2.18) can further be manipulated, we shall however leave the equation in its present form to preserve generality. Similarly, we obtain:

$$(In - Out)_v = - \int_{\Lambda_x} \int_{\Lambda_s} \int_{\partial\Lambda_v} \tilde{f}_n \dot{\mathbf{v}} \cdot d\mathbf{S}_v ds dx = - \int_{\Lambda_\psi} \nabla_v \cdot (\tilde{f}_n \dot{\mathbf{v}}) d\psi \quad (2.19)$$

and

$$(In - Out)_s = - \int_{\Lambda_x} \int_{\Lambda_v} \int_{\partial\Lambda_s} \tilde{f}_n \dot{s} \cdot d\mathbf{S}_s dv dx = - \int_{\Lambda_\psi} \partial_s (\tilde{f}_n \dot{s}) d\psi \quad (2.20)$$

for the net convective flow rate of particles entering Λ_v and Λ_s respectively through their respective boundaries $\partial\Lambda_v$ and $\partial\Lambda_s$. In the velocity and size state space, the particles move with velocities $\dot{\mathbf{v}}$ (acceleration according to Newton's second law of motion) and $\dot{s}$ (size rate of change). Here, the vectors $d\mathbf{S}_s$ and $d\mathbf{S}_v$ are the differential surface vectors respectively normal to their respective boundaries and pointing outwards. The expression $\nabla_v \cdot$ and ∂_s are the divergence operators in the three-dimensional phase space of the velocity coordinate and one-dimensional phase space of the size coordinate respectively. The net convective contribution is therefore equal to:

$$In - Out = - \int_{\Lambda_\psi} [\nabla_x \cdot (\tilde{f}_n \mathbf{v}) + \nabla_v \cdot (\tilde{f}_n \dot{\mathbf{v}}) + \partial_s (\tilde{f}_n \dot{s})] d\psi \quad (2.21)$$

Within the control volume Λ_ψ , particles can generate due to discontinuous jumps (or events) such as collisions between particles leading to aggregation and/or breakage. These

collisions can lead to change in particle velocities within a given phase space without necessarily crossing its boundaries. We express the rate of generation as:

$$Gen = \int_{\Lambda_\psi} \tilde{h}_n(\psi, t) d\psi \quad (2.22)$$

where $\tilde{h}_n$ is the number of particles generated per unit volume of particle state space and unit time owing to discrete events, mainly collisions between particles, and for this reason the generation term $\tilde{h}_n$ is also known as the collision integral. A closure hypothesis must be introduced to define this generation term in terms of the NDF because $\tilde{h}_n$ is not closed, for it involves velocity correlations between two particles (Chapman and Cowling, 1970). Putting equations (2.16), (2.21) and (2.22) together, we obtain:

$$\int_{\Lambda_\psi} [\partial_t \tilde{f}_n + \nabla_x \cdot (\tilde{f}_n \mathbf{v}) + \nabla_v \cdot (\tilde{f}_n \dot{\mathbf{v}}) + \partial_s (\tilde{f}_n \dot{s}) - \tilde{h}_n] d\psi = 0 \quad (2.23)$$

For an arbitrary control volume Λ_ψ , equation (2.23) will only hold if the integrand is zero. Thus the following relationship must hold:

$$\boxed{\partial_t \tilde{f}_n + \nabla_x \cdot (\tilde{f}_n \mathbf{v}) + \nabla_v \cdot (\tilde{f}_n \dot{\mathbf{v}}) + \partial_s (\tilde{f}_n \dot{s}) = \tilde{h}_n} \quad (2.24)$$

Equation (2.24) is the generalized population balance equation which governs the evolution of all properties describing a population of particles. Solving this equation will give information which helps characterize the system. The solution can provide knowledge on how system features such as particle size, velocity and position changes with respect to time. Having the form of an integro-differential equation and a dimensionality higher than classical transport equations, the GPBE will be extremely difficult to solve. An effective way of handling this difficulty is by using the *method of moments* which we shall describe in Section 2.4.2. This procedure turns the single GPBE into a set of transport equations with less dimensions which can be easily handled by CFD codes. It also results in loss of information but renders the problem tractable.

2.4.1 Solving the Generalized Population Balance Equation

If properly transformed, the generalized population balance equation takes the form of the Volterra or Fredholm integral equation of the second order (Ramkrishna, 2000). Solution of these forms of equations are subjects of mathematical texts (Petrovsky, 1957); so we can be certain that the GPBE can, at least in principle, be solved.

The method of *successive approximation* is one of the analytical methods from the literature. Here, the GPBE is rearranged as an integral equation in the NDF. On the left hand side of the transformed GPBE equation is an expression in terms of the NDF and on the right hand side of the equality sign an NDF functional with respect to the particle state. An approximation for the NDF of order n is substituted into the functional to find the next approximation of order $n + 1$. We refer the reader to Ramkrishna (2000) for more details. For GPBE featuring convolution integrals in the collision integral term, the *Laplace transform* is particularly suitable for obtaining analytical solutions. Ramkrishna (2000) can be referred to for an example. When the physics are simple, the *method of characteristics*

can be used to reduce the GPBE to an ordinary differential equation (ODE) along which the solution can be integrated from some known initial point. These methods fail when the physics of the system become complex.

The *method of weighted residuals* is another procedure for solving the GPBE numerically. Here, the NDF is approximated by a sum of test (or basis) functions, chosen from a linearly independent set, whose coefficients are determined so that the linear combination satisfies the GPBE. The set of test functions are derived from an orthonormal family. The coefficients are derived from an orthogonal operation on the residuals using the same set of functions used to expand the distribution or other suitable sets of weighing functions.

An alternative numerical method is the *discretized population balance approach*, also referred to as the *class method*. Here, the generalized internal state space is partitioned into classes and then the GPBE is integrated over subintervals in these classes. The GPBE is then transformed into a set of ODEs expressing macroscopic balances for the number of particles in each interval. The accuracy of the method increases with the number of classes as well as computational time. The major drawback of the method is that the number of particles is conserved only in the limit of an infinite number of classes. Also the number of discretized equations increases with the number of classes, therefore, high computational time is necessary for solving the GPBE on a sufficiently fine grid since the solution of the GPBE strongly depends on the adopted computational grid.

A more attractive approach is the *method of moments* (MOM) which we shall subsequently examine in detail. Here, special integral properties of the NDF are tracked to describe the PSD. These properties are called the *moments*. The idea is to derive transport equations for the moments of interest by integrating out all the internal coordinates from the GPBE. The number of scalars required is very small which makes implementation in CFD feasible. However, this method has been scarcely applied due to the problem of closure where moment transport equations for a given set of moments involve moments external to the set. The closure problem for the method of moments was first mentioned by Hulburt and Katz (1964) and later reviewed by Diemer and Olson (2002).

An alternative approach for solving PBEs is the Monte-Carlo simulations. This approach is mainly stochastic as specified probabilities are used to describe the particle state and random variables generated according to appropriate probabilities. The generated variables are then averaged in order to find the expected system behaviour. For more details on this approach, the reader is referred to Smith and Matsoukas (1998), Lee and Matsoukas (2000), and Rosner and Yu (2001). This technique is extremely powerful and requires a lot of computing power. With the sort of computational resources available today, it is still not practical for real applications.

2.4.2 Method of Moments

Reducing the dimensionality of the GPBE to that of classical transport equations is the idea on which the *method of moments* (MOM) of Hulburt and Katz (1964) is based. This is achieved by integrating out all the internal coordinates. In order to further illustrate this

method we shall consider a population of particles characterized by a one-dimensional particle state space and a zero collision integral. We also assume that the particles do not move. Hence, no convection in real space. In this case, equation (2.24) becomes:

$$\partial_t \tilde{f}_n + \partial_s(\tilde{f}_n \dot{s}) = 0 \quad (2.25)$$

Applying the moment transform in equation (2.11) to equation (2.25), we obtain:

$$\frac{d}{dt} \int_{\Omega_s} \tilde{f}_n s^k ds + \int_{\Omega_s} \frac{\partial}{\partial s}(\tilde{f}_n \dot{s}) s^k ds = 0 \quad (2.26)$$

Equation (2.26) is one-dimensional compared to the original GPBE in (2.25) which is two-dimensional. Only the time dependence remains when the size coordinate is integrated out. Equation (2.26) contains less information than the GPBE. Rather than a distributed information of the particle properties as obtained in the GPBE, we now obtain average values of the system properties. The method of moments therefore represents a trade-off between complexity and information available. If we represent the domain of the particle size accordingly, $\Omega_s \equiv [0, +\infty]$, equation (2.26) will yield:

$$\frac{d}{dt} \int_0^{+\infty} \tilde{f}_n s^k ds = (\tilde{f}_n \dot{s} s^k) \Big|_0 - (\tilde{f}_n \dot{s} s^k) \Big|_{+\infty} + k \int_0^{+\infty} \tilde{f}_n \dot{s} s^{k-1} ds \quad (2.27)$$

where the substitution,

$$\partial_s(\tilde{f}_n \dot{s}) s^k = \partial_s(\tilde{f}_n \dot{s} s^k) - k \tilde{f}_n \dot{s} s^{k-1} \quad (2.28)$$

have been made in equation (2.26).

According to the *regularity condition* (Chapman and Cowling, 1970), when s diverges, $\tilde{f}_n$ goes to zero faster than any other function. In our case, it means that the number of particles having infinite size is zero. The term $(\tilde{f}_n \dot{s} s^k) \Big|_{+\infty}$ on the right hand side of equation (2.27) therefore equals zero. The term $(\tilde{f}_n \dot{s} s^k) \Big|_0$ also vanishes when $s = 0$. If this be the case, then we rewrite equation (2.27) as:

$$\frac{d}{dt} \int_0^{+\infty} \tilde{f}_n s^k ds = k \int_0^{+\infty} \tilde{f}_n \dot{s} s^{k-1} ds \quad (2.29)$$

Equation (2.29) represents the moment-transport equation of the GPBE in equation (2.25). The equation (2.29) governs the evolution of the moment of order k of the NDF. We will have to write equation (2.29) z times if we wish to track the first z integer moments, with k varying from 0 to $z - 1$. The resulting set is closed if the right hand side of equation (2.29) is a function of *lower-order moments*, that is, moments of order less than k . If this is the case, additional sets of equations are not required. On the other hand, if the right hand side contains *higher-order moments*, irrespective of the number of equations we choose to consider, the resulting set remains unclosed. In order to further explain this, let us consider two different cases.

Case 1: The rate of change of particle size is a constant, $\dot{s} = s_0$. In this case, we can rewrite equation (2.29) as :

$$\frac{d}{dt} \int_0^{+\infty} \tilde{f}_n s^k ds = k s_0 \int_0^{+\infty} \tilde{f}_n s^{k-1} ds \quad (2.30)$$

Here, the problem is closed as the moment transport equation of the order k features only lower-order moments (in this case, $k - 1$).

Case 2: The rate of change of particle size is given by the power law $\dot{s} = s_0 s^\alpha$, where s_0 and α are positive constants. In this case, we can rewrite equation (2.29) as :

$$\frac{d}{dt} \int_0^{+\infty} \tilde{f}_n s^k ds = k s_0 \int_0^{+\infty} \tilde{f}_n s^{k-1+\alpha} ds \quad (2.31)$$

The right hand side of the equation (2.31) contains the moment of order $k - 1 + \alpha$ of the NDF. The resulting set is closed only when α is equal to zero or one. If α is greater than one, the equation becomes unclosed as it involves a higher-order moment.

When the method of moments leads to an unclosed set of equations, which is usually the case, closure is needed. This will require finding an approximation of high accuracy which relates the unclosed terms to the available set of moments. Subsequently, we shall present moment-closure methods in the context of the quadrature-based method of moments.

2.4.3 The Population Balance Equation (PBE) as moment transform of the GPBE

By using a similar approach to the one in section 2.4 for deriving the generalized population balance equation, we can derive the population balance equation (PBE) which has only the particle size as internal coordinate. Alternatively we can apply the method of moments in deriving the population balance equation by integrating over the velocity phase space in the generalized population balance equation. The GPBE being integro-differential and with a higher dimensionality than the PBE which we seek to derive, makes it more difficult to solve compared to the latter. We have seen in section 2.4.2 that the dimensionality of the analytical problem can be reduced at the expense of obtaining an averaged, rather than a distributed, information of the system properties.

Different moment transforms of the GPBE preserves different properties of the NDF (Mazzei, 2008). The more moments we consider, the more information we retain. Here, we shall consider the derivation of the PBE as a moment transform of the GPBE.

Let us apply the moment transform:

$$\mathcal{M}_{v,k}(\mathbf{x}, s, t) = \int_{-\infty}^{+\infty} v_1^{k_1} v_2^{k_2} v_3^{k_3} \tilde{f}_n(\mathbf{x}, s, \mathbf{v}, t) d\mathbf{v} \quad (2.32)$$

to the generalized population balance equation in (2.24). The particle velocity $\mathbf{v}$ and the particle size s are components of the generalized internal state vector $\boldsymbol{\xi}$. Equation (2.32) is the moment transform of the NDF of the k_1 -th, k_2 -th, and k_3 -th order with respect to the velocity coordinates v_1 , v_2 and v_3 respectively and of the zeroth order with respect to the size coordinate s . If we substitute in equation (2.32), $k_1 = k_2 = k_3 = 0$ and formally replace $\tilde{f}_n(\mathbf{x}, s, \mathbf{v}, t)$ with equation (2.24), we obtain:

$$\int_{-\infty}^{+\infty} [\partial_t \tilde{f}_n + \nabla_{\mathbf{x}} \cdot (\tilde{f}_n \mathbf{v}) + \nabla_{\mathbf{v}} \cdot (\tilde{f}_n \dot{\mathbf{v}}) + \partial_s (\tilde{f}_n \dot{s}) - \tilde{h}_n] d\mathbf{v} = 0 \quad (2.33)$$

The left hand side is a sum of five different integrals. The first gives :

$$\int_{-\infty}^{+\infty} \partial_t \tilde{f}_n d\mathbf{v} = \partial_t \left(\int_{-\infty}^{+\infty} \tilde{f}_n d\mathbf{v} \right) = \partial_t f_n \quad (2.34)$$

where by definition,

$$\int_{-\infty}^{+\infty} \tilde{f}_n(\mathbf{x}, s, \mathbf{v}, t) d\mathbf{v} = f_n(\mathbf{x}, s, t) \quad (2.35)$$

We can therefore define a new NDF $f_n(\mathbf{x}, s, t)$ where $f_n(\mathbf{x}, s, t) d\mathbf{x} ds$ gives the number of particles present within $d\mathbf{x}$ at the time t of interest having a size in the range ds around s .

The second integral gives:

$$\int_{-\infty}^{+\infty} \nabla_{\mathbf{x}} \cdot (\tilde{f}_n \mathbf{v}) d\mathbf{v} = \nabla_{\mathbf{x}} \cdot \int_{-\infty}^{+\infty} \tilde{f}_n \mathbf{v} d\mathbf{v} = \nabla_{\mathbf{x}} \cdot (f_n \langle \mathbf{v} | s \rangle) \quad (2.36)$$

where $\langle \mathbf{v} | s \rangle$ is the mean velocity conditioned on the particle size s .

For the third integral:

$$\int_{-\infty}^{+\infty} \nabla_{\mathbf{v}} \cdot (\tilde{f}_n \dot{\mathbf{v}}) d\mathbf{v} = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} \frac{\partial}{\partial v_i} (\tilde{f}_n \dot{v}_i) dv_1 dv_2 dv_3 \quad (2.37)$$

Here, v_i and $\dot{v}_i$ are the i -th components of the vectors $\mathbf{v}$ and $\dot{\mathbf{v}}$, respectively with respect to the Cartesian vector basis $\{\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3\}$. The limit of integration over the velocity domain extends to infinity since the Cartesian components of the velocity lies between $-\infty$ and $+\infty$. We can further decompose (2.37) thus:

$$\int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} \left[\frac{\partial}{\partial v_1} (\tilde{f}_n \dot{v}_1) + \frac{\partial}{\partial v_2} (\tilde{f}_n \dot{v}_2) + \frac{\partial}{\partial v_3} (\tilde{f}_n \dot{v}_3) \right] dv_1 dv_2 dv_3 \quad (2.38)$$

The integrands within the brackets can be evaluated separately. In the coordinate direction $\mathbf{e}_1$, we have :

$$\int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} \frac{\partial}{\partial v_1} (\tilde{f}_n \dot{v}_1) dv_1 dv_2 dv_3 = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} [\tilde{f}_n \dot{v}_1]_{-\infty}^{+\infty} dv_2 dv_3 \quad (2.39)$$

As v_1 diverges, the value of $\tilde{f}_n$ goes faster to zero than any other function. This is the regularity condition (refer to Section 2.3.2). Therefore,

$$[\tilde{f}_n \dot{v}_1]_{-\infty}^{+\infty} = 0 \quad (2.40)$$

This same result is true for the other two coordinate directions; hence, the integral in equation (2.37) is zero. We arrive at the same result if we use a different approach: the Gauss theorem. Applying the Gauss theorem in the velocity state space:

$$\int_{-\infty}^{+\infty} \nabla_{\mathbf{v}} \cdot (\tilde{f}_n \dot{\mathbf{v}}) d\mathbf{v} = \int_{-\infty}^{+\infty} \tilde{f}_n \dot{\mathbf{v}} \cdot d\mathbf{S}_{\mathbf{v}} = 0 \quad (2.41)$$

As earlier said, the regularity condition on the boundary $\partial\Omega_v$ has the same effect (Ω_v stretches out to infinity).

The fourth integral in the equation (2.33) gives:

$$\int_{-\infty}^{+\infty} \partial_s(\tilde{f}_n \dot{s}) d\mathbf{v} = \partial_s \left(\int_{-\infty}^{+\infty} \dot{s} \tilde{f}_n d\mathbf{v} \right) = \partial_s(f_n \langle \dot{s} | s \rangle) \quad (2.42)$$

where $\langle \dot{s} | s \rangle$ is the mean velocity in size space conditioned on the particle size s .

And the fifth term can be written thus:

$$\int_{-\infty}^{+\infty} \tilde{h}_n d\mathbf{v} \equiv h_n \quad (2.43)$$

If we combine all the results obtained, we get:

$$\partial_t f_n + \nabla_{\mathbf{x}} \cdot (f_n \langle \mathbf{v} | s \rangle) + \partial_s(f_n \langle \dot{s} | s \rangle) - h_n = 0 \quad (2.44)$$

Re-arranging (2.44), we obtain:

$$\boxed{\partial_t f_n + \nabla_{\mathbf{x}} \cdot (f_n \langle \mathbf{v} | s \rangle) + \partial_s(f_n \langle \dot{s} | s \rangle) = h_n} \quad (2.45)$$

Equation (2.45) is the population balance equation. In the absence of growth in size space the population balance reduces to:

$$\boxed{\partial_t f_n + \nabla_{\mathbf{x}} \cdot (f_n \langle \mathbf{v} | s \rangle) = h_n} \quad (2.46)$$

Note: During the course of this work we will switch between the size-based NDF $f_n(\mathbf{x}, s, t)$ and the volume-based NDF $\hat{f}_n(\mathbf{x}, v_p, t)$. For simplicity, we use the size-based NDF for most of our derivations and subsequently introduce the volume-based equivalent when needed.

2.5 Closures based on Quadrature Approximations

In order to simplify the mathematics and clearly describe the methodology for closure, we shall limit our discussion to the univariate form of the generalized population balance derived in the previous section. If we were to consider the GPBE in section 2.4, there are three possible scenarios.

Scenario 1: The particles are *motionless*: In this case, the internal state space of the population is fully described by the size coordinate and the system is purely univariate as the velocity coordinate is absent.

Scenario 2: Particles *move at the same velocity*: In this case, we assume the particles move at the same velocity as the fluid if the flow is laminar. For turbulent flows, we assume a relationship for the particle velocity as a function of time and space.

Scenario 3: The particle *velocity is a function of the particle size*: Here, particles within

a particular differential size range move with a certain velocity different from that of particles external to the set. This is similar to the PBE derived in section 2.4.3.

For the time being, we shall limit our discussion to the univariate PBE derived in the previous section. From the PBE, the function $f_n(\mathbf{x}, s, t)$ describes the distribution of the population over the size coordinate s . We also assume that the velocity $\langle \mathbf{v} | s \rangle$ of the particles is an average velocity which is known if particle size is known. Therefore, the NDF needs not describe the distribution over the velocity. This is because if the number density of elements with the internal coordinate in the range ds around s is $f_n(\mathbf{x}, s, t)ds$, the same is the number of particles with velocity in the range $d\mathbf{v}$ around $\mathbf{v}(\mathbf{x}, s, t)$. Moreover, the number of particles with size and velocity external to the range specified is clearly zero.

To overcome the problem of closure described in section 2.4.2, the size-based NDF is replaced by a finite-mode approximation given by a summation of dirac delta functions:

$$f_n(\mathbf{x}, s, t) \approx \sum_{i=1}^{\nu} n_i(\mathbf{x}, t) \delta[s - s_i(\mathbf{x}, t)] \quad (2.47)$$

The formula in equation (2.47) is the quadrature approximation referring to a univariate distribution. For a multivariate distribution, the method approximates the NDF as a summation of multidimensional dirac delta functions. The monovariate distribution represented in (2.47) represents the disperse phase by ν different classes each having a number density $n_i(\mathbf{x}, t)$ and a particle size $s_i(\mathbf{x}, t)$. The scalar functions $n_i(\mathbf{x}, t)$ and $s_i(\mathbf{x}, t)$ are the weights and nodes of the quadrature respectively.

The quadrature formula eliminates the closure problem. It is a common factor in the QMOM and DQMOM methods with the difference in the two methods being the way in which the nodes and weights are computed. We shall first present the direct quadrature method of moments, considering scenario 3 above where the particle velocity is conditioned on the particle size.

2.5.1 The Direct Quadrature Method of Moments (DQMOM)

For a univariate distribution, the PBE is represented by equation (2.45):

$$\partial_t f_n + \nabla_{\mathbf{x}} \cdot (f_n \langle \mathbf{v} | s \rangle) + \partial_s (f_n \langle \dot{s} | s \rangle) = h_n$$

We shall analyse the DQMOM technique in steps to guide the reader through the theoretical analysis of the method.

Step 1 : *Substitute the quadrature approximation into the PBE.*

We adopt the quadrature formula of equation (2.47) for the PBE by representing the NDF by ν different classes of particle each having a number density $n_i(\mathbf{x}, t)$ and a particle size $s_i(\mathbf{x}, t)$.

The first term of the PBE then becomes:

$$\partial_t f_n = \partial_t \sum_{i=1}^{\nu} n_i \delta(s - s_i) = \sum_{i=1}^{\nu} [\partial_t n_i \delta(s - s_i) - n_i \partial_t s_i \delta'(s - s_i)] \quad (2.48)$$

where the product chain rule has been applied. The function described by the expression $\delta'(s - s_i)$ represents the derivative of $\delta(s - s_i)$ with respect to $(s - s_i)$ and further defined by the integral property (Arfken and Weber, 1985):

$$\int_{s_a}^{s_c} f(s) \delta'(s - s_b) ds = -\partial_s f(s_b) \quad ; \quad s_a < s_b < s_c \quad (2.49)$$

In deriving transport equations from the GPBE, this property becomes very useful as we must integrate the latter and compute its moments. If we substitute the identity:

$$n_i \partial_t s_i = \partial_t (n_i s_i) - s_i \partial_t n_i \quad (2.50)$$

in equation (2.48), we obtain:

$$\partial_t f_n = \sum_{i=1}^v [\delta \partial_t n_i - \delta' \partial_t (n_i s_i) + \delta' s_i \partial_t n_i] \quad (2.51)$$

The second term of the PBE can be manipulated in a similar way. If there exist a conditional relationship between the particle velocity and the particle size, then v is a known function of the variables $s, \mathbf{x}, t$; accordingly, we can write:

$$\begin{aligned} f_n \langle v | s \rangle &= \left(\sum_{i=1}^v n_i(\mathbf{x}, t) \delta[s - s_i(\mathbf{x}, t)] \right) v(\mathbf{x}, s, t) \\ &= \sum_{i=1}^v n_i(\mathbf{x}, t) v[\mathbf{x}, s_i(\mathbf{x}, t), t] \delta[s - s_i(\mathbf{x}, t)] \\ &= \sum_{i=1}^v n_i(\mathbf{x}, t) v_i(\mathbf{x}, t) \delta[s - s_i(\mathbf{x}, t)] \end{aligned} \quad (2.52)$$

where v_i is the velocity of the particles of the quadrature class i . Thus:

$$\nabla_{\mathbf{x}} \cdot (f_n \langle v | s \rangle) = \nabla_{\mathbf{x}} \cdot \sum_{i=1}^v n_i v_i \delta = \sum_{i=1}^v (\nabla_{\mathbf{x}} \cdot (n_i v_i) \delta - n_i v_i \cdot \nabla_{\mathbf{x}} s_i \delta') \quad (2.53)$$

But:

$$n_i v_i \cdot \nabla_{\mathbf{x}} s_i = \nabla_{\mathbf{x}} \cdot (n_i s_i v_i) - s_i \nabla_{\mathbf{x}} \cdot (n_i v_i) \quad (2.54)$$

Therefore, we obtain:

$$\nabla_{\mathbf{x}} \cdot (f_n \langle v | s \rangle) = \sum_{i=1}^v (\nabla_{\mathbf{x}} \cdot (n_i v_i) \delta - \nabla_{\mathbf{x}} \cdot (n_i s_i v_i) \delta' + s_i \nabla_{\mathbf{x}} \cdot (n_i v_i) \delta') \quad (2.55)$$

The third term which represents convection in size space can be treated in a similar way as the second term. If there exist a conditional relationship between the particle growth rate and particle size, then $\dot{s}$ is a known function of the variables $s, \mathbf{x}, t$; accordingly, we can write:

$$\begin{aligned} f_n \langle \dot{s} | s \rangle &= \left(\sum_{i=1}^v n_i(\mathbf{x}, t) \delta[s - s_i(\mathbf{x}, t)] \right) \dot{s}(\mathbf{x}, s, t) \\ &= \sum_{i=1}^v n_i(\mathbf{x}, t) \dot{s}[\mathbf{x}, s_i(\mathbf{x}, t), t] \delta[s - s_i(\mathbf{x}, t)] \\ &= \sum_{i=1}^v n_i(\mathbf{x}, t) \dot{s}_i(\mathbf{x}, t) \delta[s - s_i(\mathbf{x}, t)] \end{aligned} \quad (2.56)$$

where $\dot{s}_i$ is the growth rate of the particles of the quadrature class i . Thus:

$$\partial_s(f_n \dot{s}) = \partial_s \left(\sum_{i=1}^v n_i \dot{s}_i \delta \right) = \sum_{i=1}^v n_i \dot{s}_i \delta' \quad (2.57)$$

Putting all the results together, we obtain:

$$\sum_{i=1}^v (c_i^n \delta - (c_i^s - s_i c_i^n) \delta' + n_i \dot{s}_i \delta') = h_n \quad (2.58)$$

where by definition,

$$c_i^n(\mathbf{x}, t) \equiv \partial_t n_i + \nabla_{\mathbf{x}} \cdot (n_i \mathbf{v}_i) \quad ; \quad c_i^s(\mathbf{x}, t) \equiv \partial_t (n_i s_i) + \nabla_{\mathbf{x}} \cdot (n_i s_i \mathbf{v}_i) \quad (2.59)$$

Equation (2.58) expresses the PBE for a univariate distribution of particles whose velocities is dependent on the particle size and whose NDF fulfils the equation (2.47). The unknowns are the weights $n_i(\mathbf{x}, t)$ and nodes $s_i(\mathbf{x}, t)$ of the quadrature approximation. Equation (2.59) can be regarded as transport equations for the weights and weighted nodes as they help govern their evolution in time and space. In order to find the source terms $c_i^n(\mathbf{x}, t)$ and $c_i^s(\mathbf{x}, t)$ in (2.59), we have to compute the moment transforms of the discretized PBE.

Step 2 : Determine the source terms via moment transform.

In order to determine the source terms, we take the moment transform of Eq.(2.58). Equation (2.58) has $2v$ unknowns, therefore an equivalent number of moments has to be considered. Different moments preserve different properties of the NDF. Hence, the source term depends on the moments selected.

We shall denote the moment transform of the NDF by $\mathcal{M}_k(\mathbf{x}, t)$. For the first term of the equation (2.58), we have:

$$\mathcal{M}_k \left(\sum_{i=1}^v c_i^n \delta \right) = \sum_{i=1}^v c_i^n \int_0^\infty s^k \delta ds = \sum_{i=1}^v c_i^n s_i^k \quad (2.60)$$

Applying the integral property defined in (2.49), the second term of equation (2.58) becomes:

$$\mathcal{M}_k \left(\sum_{i=1}^v (c_i^s - s_i c_i^n) \delta' \right) = \sum_{i=1}^v (c_i^s - s_i c_i^n) \int_0^\infty s^k \delta' ds = -k \sum_{i=1}^v (c_i^s - s_i c_i^n) s_i^{k-1} \quad (2.61)$$

In like manner, the third term becomes :

$$\mathcal{M}_k \left(\sum_{i=1}^v n_i \dot{s}_i \delta' \right) = \sum_{i=1}^v n_i \dot{s}_i \int_0^\infty s^k \delta' ds = -k \sum_{i=1}^v n_i s_i^{k-1} \dot{s}_i \quad (2.62)$$

Putting all the results together , we obtain :

$$(1 - k) \sum_{i=1}^v c_i^n s_i^k + k \sum_{i=1}^v c_i^s s_i^{k-1} = \mathcal{M}_k(h_n) + k \sum_{i=1}^v n_i s_i^{k-1} \dot{s}_i \quad (2.63)$$

Equation (2.63) is an algebraic equation in $2v$ unknowns: $c_i^n(\mathbf{x}, t)$ and $c_i^s(\mathbf{x}, t)$. In order to solve for these unknowns, we need $2v$ independent set of equations. To obtain a linear system of $2v$ independent equations we write equation (2.63) for $2v$ independent set of moments of the PBE. The best choice of moments will depend on the problem at hand. In this work, we use the first $2v$ integer moments of the PBE because of their significance to the problem. For example, the zeroth order size based moment represents the total number of particles per unit physical volume while the second and third order moments are related to the total particle area and the total particle volume respectively. Although fractional-order moments ($k \in \mathbb{R}$) can be adopted, here, we use integer-order moments ($k \in \mathbb{N}$).

The linear system of equations derived from using the first $2v$ moments can be represented as a set of matrices in the form $\mathbf{X}\mathbf{c} = \mathbf{z}$ where $\mathbf{X}$ is a $2v \times 2v$ matrix, $\mathbf{c}$ and $\mathbf{z}$ are both column matrices. If $x_{p,q}$ are the elements of $\mathbf{X}$, then the elements of the matrix are:

$$\begin{aligned} x_{p,q} &= (2-p)s_q^{p-1} & ; & & 1 \leq p \leq 2v & ; & & 1 \leq q \leq v \\ x_{p,q} &= (p-1)s_q^{p-2} & ; & & 1 \leq p \leq 2v & ; & & v+1 \leq q \leq 2v \end{aligned} \quad (2.64)$$

The elements c_p of the column vector $\mathbf{c}$ are:

$$\begin{aligned} c_p &= c_p^n & ; & & 1 \leq p \leq v \\ c_p &= c_p^s & ; & & v+1 \leq p \leq 2v \end{aligned} \quad (2.65)$$

and the elements z_p of the column vector $\mathbf{z}$ are:

$$z_p = \mathcal{M}_{p-1}(h_n) + (p-1) \sum_{i=1}^v n_i s_i^{p-2} \dot{s}_i \quad ; \quad 1 \leq p \leq 2v \quad (2.66)$$

If the determinant of $\mathbf{X}$ is non-zero, then we can obtain $c_i^n(\mathbf{x}, t)$ and $c_i^s(\mathbf{x}, t)$ by simply inverting $\mathbf{X}$. Sometimes, the nodes are not unique and the matrix $\mathbf{X}$ may not be full rank making it impossible to invert $\mathbf{X}$. This may be due to a redundant subset of the v delta functions representing the NDF. To overcome this, a little perturbation is introduced to the indistinct node so that $\mathbf{X}$ becomes full rank. The perturbations introduced in $\mathbf{X}$ leaves the weights $n_i(\mathbf{x}, t)$ unchanged (Marchisio and Fox, 2005).

The $2v$ transport equations from equation (2.59) that govern the evolution of the weights and weighted nodes in time and space are now closed provided that the collision integral h_n and the particle size growth rate $\dot{s}$ are known. The theory of transport phenomena provides constitutive expressions that can describe the growth rate for the particle size. The reader is referred to Levenspiel (1999) for more on this subject. Modelling the collision integral h_n is the subject of the next chapter.

Step 3: Assign Boundary and Initial Conditions.

We must assign both boundary and initial conditions in order to solve the problem in equation (2.59). To initialize the quadrature weights and weighted nodes the initial values of at least $2v$ moments of the NDF has to be known. If we plug these known moments into

a discretized moment equation, we obtain a set of non-linear algebraic equations whose solutions yield $n_i(\mathbf{x}, t)$ and $s_i(\mathbf{x}, t)$. On the boundaries of the system, we can specify any one of the following: the weights, the weighted nodes or their fluxes.

We can assign the initial conditions if we know at least 2ν independent moments of the NDF at time t_0 . If we assume to know the function $f_n(\mathbf{x}, s, t)$, we can calculate its moments as:

$$\mathcal{M}_k(\mathbf{x}, t_0) = \int_0^\infty f_n(\mathbf{x}, s, t_0) s^k ds \quad (2.67)$$

Using equation (2.47), we can write:

$$\mathcal{M}_k(\mathbf{x}, t_0) = \sum_{i=1}^v n_i(\mathbf{x}, t_0) s_i^k(\mathbf{x}, t_0) \quad (2.68)$$

Equation (2.68) is a non-linear equation in 2ν unknowns $n_i(\mathbf{x}, t_0)$ and $s_i(\mathbf{x}, t_0)$. For any value of ν the equation has 2ν unknowns which can be found by writing and solving 2ν equations for 2ν different values of k . Any set of moments can be chosen. However, the resulting algebraic system and initial conditions will depend largely on our choice of k .

Step 4: *Solve the resulting non-linear algebraic system using an appropriate method.*

For a univariate system, the solution for the resulting algebraic system will be based on the product-difference algorithm by Gordon (1968). This method uses the 2ν NDF moments to construct a real symmetric tridiagonal matrix whose eigenvalues coincide with the quadrature nodes and the eigenvectors allow for the determination of the quadrature weights.

Say we choose the first 2ν moments of the NDF, then it follows that:

$$\int_0^\infty f_n(\mathbf{x}, s, t_0) s^k ds = \sum_{i=1}^v n_i(\mathbf{x}, t_0) s_i^k(\mathbf{x}, t_0) \quad (2.69)$$

for any integer value of k from 0 to $2\nu - 1$. Hence, it must also be:

$$\int_0^\infty f_n(\mathbf{x}, s, t_0) p_k(s) ds = \sum_{i=1}^v n_i(\mathbf{x}, t_0) p_k[s_i(\mathbf{x}, t_0)] \quad (2.70)$$

for any polynomial $p_k(s)$ of degree k with $0 \leq k \leq 2\nu - 1$. This holds if the integration is distributive and $p_k(s)$ is a linear combination of polynomials s^k with $0 \leq k \leq 2\nu - 1$. If this is the case, then the choice of nodes and weights makes equation (2.47) a *Gaussian quadrature*. Instead of an accuracy of $\nu - 1$ yielded by a normal quadrature formula for ν nodes, the Gaussian quadrature reaches an accuracy of order $2\nu - 1$. If we had made a different choice of moments in equation (2.68) other than the first 2ν moments, equation (2.47) would no longer be a Gaussian quadrature because our choice would lead to different nodes and weights. Therefore, it is important from the standpoint of mathematical accuracy that we preserve the first 2ν moments of the NDF.

This important property suggests an efficient alternative of calculating the quadrature coefficients $s_i(\mathbf{x}, t_0)$. These, in fact, are the roots of the monic polynomial $\pi_v(\mathbf{x}, s, t_0)$ of degree v based on the inner product definition:

$$\langle p_i, p_j \rangle \equiv \int_0^\infty f_n(\mathbf{x}, s, t) p_i(s) p_j(s) ds \quad ; \quad \forall p_i, p_j \in \mathbb{P}_n(s) \quad (2.71)$$

that adopts as weighing function the NDF itself. $f_n(\mathbf{x}, s, t)$ is positive-defined as required by the inner product definition. Here, $p_i(s)$ and $p_j(s)$ are polynomials of the vector space $\mathbb{P}_n(s)$ of real polynomials $p(s)$ of degree at most n .

We can find the nodes by constructing a sequence of monic orthogonal polynomials by relating any three consecutive polynomial in the following sequence:

$$\pi_{k+1}(s) = (s - \alpha_k) \pi_k(s) - \beta_k \pi_{k-1}(s) \quad (2.72)$$

where the following definition holds:

$$\pi_{-1}(s) \equiv 0 \quad ; \quad \pi_0(s) \equiv 1 \quad ; \quad \alpha_k \equiv \frac{\langle \pi_k, s \pi_k \rangle}{\langle \pi_k, \pi_k \rangle} \quad ; \quad \beta_k \equiv \frac{\langle \pi_k, \pi_k \rangle}{\langle \pi_{k-1}, \pi_{k-1} \rangle} > 0 \quad (2.73)$$

(2.72) is the recursive relation. This relationship can be re-arranged thus:

$$s \pi_k(s) = \alpha_k \pi_k(s) + \beta_k \pi_{k-1}(s) + \pi_{k+1}(s) \quad (2.74)$$

Combining equations (2.73) and (2.74) helps us generate a linear system of polynomials whose roots provide the abscissas $s_i(\mathbf{x}, t_0)$. Although finding the roots of these set of polynomials is not a trivial task as the problem is often ill-conditioned (Press et al., 2002) and numerically do not converge easily to give a solution, the product-difference algorithm of Gordon (1968) makes all the difference by solving an eigenproblem involving a real symmetric tridiagonal matrix to compute the roots of $\pi_v(\mathbf{x}, s, t_0)$ and simultaneously the weights and nodes. The linear system generated is given as :

$$s \pi_{v-1}(s) = \mathbf{A} \pi_{v-1}(s) + \mathbf{r}_v(s) \quad (2.75)$$

Where $\mathbf{A}$ is a matrix of the coefficients of the polynomial given by:

$$\mathbf{A} = \begin{pmatrix} \alpha_0 & 1 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 & 0 & 0 \\ \beta_1 & \alpha_1 & 1 & \cdots & 0 & 0 & 0 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \cdots & \vdots & \vdots & \vdots & \cdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & \beta_k & \alpha_k & 1 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \cdots & \vdots & \vdots & \vdots & \cdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & \beta_{v-2} & \alpha_{v-2} & 1 \\ 0 & 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 & \beta_{v-1} & \alpha_{v-1} \end{pmatrix} \quad (2.76)$$

The vector $\pi_{v-1}(s)$ of the polynomials $\pi_k(s)$ of degree k smaller than v is:

$$\pi_{v-1}(s) = (\pi_0 \quad \pi_1 \quad \pi_2 \quad \cdots \quad \pi_k \quad \cdots \quad \pi_{v-2} \quad \pi_{v-1})^T \quad (2.77)$$

and the residual vector $\mathbf{r}_v(s)$ is:

$$\mathbf{r}_v(s) = (0 \ 0 \ 0 \ \cdots \ 0 \ 0 \ \cdots \ \pi_v)^T \quad (2.78)$$

The quadrature nodes are the roots of the polynomial $\pi_v(s)$, for any $s_i(x, t_0)$ the vector $\mathbf{r}_v(s_i)$ vanishes and the (2.75) reduces to:

$$s_i \pi_{v-1}(s_i) = \mathbf{A} \pi_{v-1}(s_i) \quad (2.79)$$

We can say from (2.79) that *the eigenvalues of the matrix $\mathbf{A}$ coincide with the quadrature nodes $s_i(x, t_0)$* . In our case, $\mathbf{A}$ is an unsymmetrical real tridiagonal matrix. It is therefore convenient to transform the matrix $\mathbf{A}$ into a symmetric one having same eigenvalues. To do this we refer to Lipschutz and Lipson (2001) and carry out a diagonalization procedure by finding an invertible matrix $\mathbf{D}$ that satisfies the similarity condition:

$$\mathbf{B} = \mathbf{D} \mathbf{A} \mathbf{D}^{-1} \quad (2.80)$$

where $\mathbf{B}$ is a real symmetric matrix. If such a matrix as $\mathbf{D}$ exists, then $\mathbf{A}$ and $\mathbf{B}$ are similar having the same eigenvalues and characteristic equation. On the other hand, their eigenvectors differ and are related thus:

$$\boldsymbol{\tau}(s_i) = \mathbf{D}^{-1} \pi_{v-1}(s_i) \quad (2.81)$$

where $\boldsymbol{\tau}(s_i)$ is the eigenvector of $\mathbf{B}$ for the value of the eigenvalue corresponding to $s_i(x, t_0)$. For tridiagonal matrices the transformation in (2.80) is always possible and $\mathbf{D}$ is diagonal with diagonal elements d_k given by:

$$d_1 = 1 \quad ; \quad d_k = \left(\prod_{i=1}^{k-1} \sqrt{\beta_i} \right)^{-1} \quad ; \quad 2 \leq k \leq v \quad (2.82)$$

On performing the transformation in (2.78), we obtain:

$$\mathbf{B} = \begin{pmatrix} \alpha_0 & \sqrt{\beta_1} & \cdots & 0 & 0 & 0 & \cdots & 0 & 0 \\ \sqrt{\beta_1} & \alpha_1 & \cdots & 0 & 0 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \cdots & \vdots & \vdots & \vdots & \cdots & \vdots & \vdots \\ 0 & 0 & \cdots & \sqrt{\beta_k} & \alpha_k & \sqrt{\beta_{k+1}} & \cdots & 0 & 0 \\ \vdots & \vdots & \cdots & \vdots & \vdots & \vdots & \cdots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & \alpha_{v-2} & \sqrt{\beta_{v-1}} \\ 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & \sqrt{\beta_{v-1}} & \alpha_{v-1} \end{pmatrix} \quad (2.83)$$

We therefore have a new eigenproblem which enables us to solve for the quadrature nodes:

$$\mathbf{B} \boldsymbol{\tau}(s_i) = s_i \boldsymbol{\tau}(s_i) \quad (2.84)$$

In order to proceed with solving for the eigenvalues of $\mathbf{B}$, the elements of the matrix itself, α_k and β_k , have to be evaluated using the recursive relation of equations (2.72) and (2.73). This could be a very laborious task to accomplish. Therefore we turn to a more effective method: the product-difference algorithm of Gordon (1968). The algorithm is

as follows: First, the elements $g_{p,q}$ of the *Gordon matrix* $\mathbf{G}$, a square matrix of dimension $2v+1$, must be determined. The elements of the first column of the matrix are given by:

$$g_{1,1} = 1 \quad ; \quad g_{p,1} = 0 \quad ; \quad 2 \leq p \leq 2v+1 \quad (2.85)$$

The elements of the second column are:

$$g_{p,2} = \mathcal{M}_{p-1}(\mathbf{x}, t_0)(-1)^{p-1} \quad ; \quad 1 \leq p \leq 2v \quad ; \quad g_{p,2v+1} = 0 \quad (2.86)$$

The remaining elements are found using the product-difference algorithm:

$$g_{p,q} = g_{1,q-1} g_{p+1,q-2} - g_{1,q-2} g_{p+1,q-1} ; \quad \begin{cases} 1 \leq p \leq 2(v+1) - q \\ 3 \leq q \leq 2v+1 \end{cases} \quad (2.87)$$

$$g_{p,q} = 0 ; \quad \begin{cases} 2(v+1) - (q-1) \leq p \leq 2v+1 \\ 3 \leq q \leq 2v+1 \end{cases}$$

The coefficients α_k and β_k are then given by :

$$\begin{aligned} \alpha_k &= \varsigma_{2k} + \varsigma_{2k+1} & ; & & 0 \leq k \leq v-1 \\ \beta_k &= \varsigma_{2k} \varsigma_{2k-1} & ; & & 1 \leq k \leq v-1 \end{aligned} \quad (2.88)$$

where ς_k is defined as:

$$\varsigma_0 = 0 \quad ; \quad \varsigma_k = \frac{g_{1,k+2}}{g_{1,k+1} g_{1,k}} \quad ; \quad 1 \leq k \leq 2v-1 \quad (2.89)$$

Once the elements of $\mathbf{B}$ are determined, the eigenvalues and eigenvectors can be found. The eigenvector is needed to compute the quadrature weights. The quadrature weight can be derived from the product-difference scheme as follows:

$$n_i(\mathbf{x}, t_0) = \mathcal{M}(\mathbf{x}, t_0) \tau_1^2(\mathbf{x}, s_i, t_0) \quad (2.90)$$

where $\tau_1(\mathbf{x}, s_i, t_0)$ is the first component of the eigenvector $\boldsymbol{\tau}(\mathbf{x}, s_i, t_0)$ of matrix $\mathbf{B}$. In equation (2.90), the eigenvectors are assumed of unit magnitude. Note that in general the eigenvectors are defined up to a proportionality constant, therefore only ratios between their components and not the components are meaningful (Mazzei, 2008).

2.5.2 Quadrature Method of Moments (QMOM)

In order to track the moments of the NDF, the quadrature method of moments back-calculates the quadrature nodes and weights by solving the system of non-linear algebraic equations generated from equation (2.68). For univariate distributions, the product-difference algorithm described in section 2.5.1 can similarly be applied to obtain a solution. However, the main difference between the DQMOM and QMOM methods is that whereas the former runs the algorithm once to initialize the quadrature variables, the latter runs it at every time step of the numerical simulation and at each grid point of the computational domain.

To derive the transport equations of the NDF moments, we use a similar transformation as described in Section 2.3.3. Let us recall the PBE, where a conditional relationship exists between the particle velocity and the particle size:

$$\partial_t f_n + \nabla_{\mathbf{x}} \cdot (f_n \langle \mathbf{v} | s \rangle) + \partial_s (f_n \langle \dot{s} | s \rangle) = h_n$$

Let us apply the moment transform to the PBE. For the first term, we write:

$$\mathcal{M}_k(\partial_t f_n) = \int_0^\infty (\partial_t f_n) s^k ds = \int_0^\infty \partial_t (f_n s^k) ds = \partial_t \int_0^\infty f_n s^k ds = \partial_t \mathcal{M}_k \quad (2.91)$$

For the second term, we write :

$$\mathcal{M}_k[\nabla_{\mathbf{x}} \cdot (f_n \mathbf{v})] = \int_0^\infty \nabla_{\mathbf{x}} \cdot (f_n \mathbf{v}) s^k ds = \int_0^\infty \nabla_{\mathbf{x}} \cdot (f_n s^k \mathbf{v}) ds \quad (2.92)$$

$$= \nabla_{\mathbf{x}} \cdot \int_0^\infty f_n s^k \mathbf{v} ds = \nabla_{\mathbf{x}} \cdot (\mathcal{M}_k \mathbf{v}_k) \quad (2.93)$$

where the velocity $\mathbf{v}_k(\mathbf{x}, t)$ with which the k -th moment is convected is a mean velocity defined by:

$$\mathbf{v}_k(\mathbf{x}, t) \equiv \frac{1}{\mathcal{M}_k(\mathbf{x}, t)} \int_0^\infty f_n(\mathbf{x}, s, t) s^k \mathbf{v}(\mathbf{x}, s, t) ds \quad (2.94)$$

Here, $\mathbf{v}(\mathbf{x}, s, t)$ is the particle mean velocity conditioned on the particle size. Applying the quadrature approximation, we have:

$$f_n(\mathbf{x}, s, t) s^k \mathbf{v}(\mathbf{x}, s, t) = \sum_{i=1}^v n_i(\mathbf{x}, t) s_i^k(\mathbf{x}, t) \mathbf{v}_i(\mathbf{x}, t) \delta[s - s_i(\mathbf{x}, t)] \quad (2.95)$$

substituting the relation (2.95) into equation (2.94) we have :

$$\mathbf{v}_k(\mathbf{x}, t) = \frac{1}{\mathcal{M}_k(\mathbf{x}, t)} \sum_{i=1}^v n_i(\mathbf{x}, t) s_i^k(\mathbf{x}, t) \mathbf{v}_i(\mathbf{x}, t) \quad (2.96)$$

We have previously evaluated the moments of the third and fourth term. The reader is referred to equations (2.57), (2.62) and (2.63). If we make all necessary substitutions, we obtain:

$$\partial_t \mathcal{M}_k + \nabla_{\mathbf{x}} \cdot (\mathcal{M}_k \mathbf{v}_k) = \mathcal{S}_k(\mathbf{x}, t) \quad (2.97)$$

where $\mathcal{S}_k$ is the source term given by:

$$\mathcal{S}_k \equiv \mathcal{M}_k(h_n) + k \sum_{i=1}^v n_i s_i^{k-1} \dot{s}_i \quad (2.98)$$

The relationship in Eq. (2.97) governs the evolution of the moment of the NDF, $\mathcal{M}_k(\mathbf{x}, t)$. The quadrature nodes and weights are the functions of real interest which can be back-calculated from the product-difference algorithm. We have $2v$ unknowns which will require $2v$ equations to solve. As before we refer to equation (2.68), which relates the nodes, weights and the moments making it possible for $2v$ moments to be tracked. So, in this

case, for each time step of the numerical simulation and at each grid point of the computational domain, the Gordon matrix $\mathbf{G}$ can be generated once the moments are determined. The matrix $\mathbf{G}$ is generated using the equations (2.85), (2.86) and (2.87). Consequently, a tridiagonal symmetric matrix $\mathbf{B}$ can be built and its eigenvalues and eigenvectors computed. These further gives results for the quadrature weights $n_i(\mathbf{x}, t)$ and nodes $s_i(\mathbf{x}, t)$.

In theory we can say QMOM and DQMOM are similar, but numerically, the latter poses several advantages over the former: QMOM is computationally expensive as it has to run the PD algorithm for each time step, track the NDF and then back-calculate to determine the functions of interest whereas for DQMOM, the quadrature nodes and weights are tracked directly. Also, the moment equations are not as numerically stable as those involving the nodes and weights. This makes DQMOM a preferred method and the method of choice when dealing with multivariate distributions.

Chapter 3

Mathematical Modelling of Aggregation and Breakage

In this chapter we discuss the principal mathematical theories underlying the modelling of birth and death functions due to aggregation and breakage in gas-solid fluidized beds. In an attempt to close the generation term h_n encountered in chapter two:

- 1) we shall show a detailed derivation of the breakage and aggregation equations;*
- 2) we shall discuss a few modelling aspects to the aggregation and breakage kernel.*

3.1 Introduction

Most chemical and mechanical processes involve change in size and shape of the dispersed phase or solid matter. Changes in size can be as a result of particle breakage (or size reduction) and/or aggregation (or size enlargement). In fluidized beds, these phenomena (breakage and aggregation) contribute a great deal to the eventual properties of the system and should be accounted for while modelling. We account for aggregation and breakage in the generation term h_n of the population balance equation.

We can write equations to describe aggregation and breakage in the disperse phase and incorporate same in the PBE. The aggregation and breakage equations contain grouped terms in their expressions referred to as kernels. These kernels are major parameters in the aggregation and breakage equations.

There are three basic approaches to modelling these kernels: theoretical, empirical and experimental (Narni et al., 2012). Smoluchowski (1918) derived the shear and brownian aggregation kernel via theoretical considerations whereas Sastry (1975) and Kapur (1972) employed empirical approach to obtain their kernels. Peglow (2005) fitted experimental data to modify the empirical kernel proposed by Kapur (1972). Recently, due to the advancement of computational methods, some authors have extracted kernels using computer simulations (Tan et al., 2004; Gantt et al., 2006).

These kernels are usually expressed as a product of two terms: the collision frequency and the collision efficiency. In this work, we shall try to select or develop appropriate kernels which can help to best describe a dense polydisperse fluidized bed.

3.2 Breakage and Aggregation Equations

Aggregation and breakage play an important role in several chemical processes: precipitation, crystallization, separation, granulation and reaction in multiphase systems (Marchisio et al., 2003a). The evolution of the PSD is not fully described if the contributions of aggregation and breakage are not considered.

Particles generate as a result of collisions. If we no longer regard the velocity as an internal coordinate, we will need to account for this effect indirectly. Collisions make particles aggregate and break; we can consequently write:

$$h_n(\mathbf{x}, s, t) = h_n^b(\mathbf{x}, s, t) + h_n^a(\mathbf{x}, s, t) \quad (3.1)$$

where $h_n^b(\mathbf{x}, s, t)$ and $h_n^a(\mathbf{x}, s, t)$ are the breakage and aggregation source terms respectively.

3.2.1 The Breakage Equation

The generation term h_n in the PBE in part accounts for breakage. The first step in closing the breakage term is to define the source and sink terms due to breakage. Consequently we can write:

$$h_n^b(\mathbf{x}, s, t) = B_b(\mathbf{x}, s, t) - D_b(\mathbf{x}, s, t) \quad (3.2)$$

where

$B_b(\mathbf{x}, s, t)$ = the rate of birth of particles of size s due to breakage; and
 $D_b(\mathbf{x}, s, t)$ = the rate of death of particles of size s due to breakage.

Consider a differential volume $d\mathbf{x}$ around $\mathbf{x}$ in physical space $\Omega_{\mathbf{x}}$ and dr around r in size space Ω_s . If we assume that particles break up independent of each other and assign $b(\mathbf{x}, r, t)$ to be the fraction of the number of particles with size range dr around r per unit volume of physical space breaking per unit time, we can define the number of particles of size range dr around r breaking per unit time per unit volume of physical space as:

$$b(\mathbf{x}, r, t) f_n(\mathbf{x}, r, t) dr \quad (3.3)$$

where $b(\mathbf{x}, r, t)$ is also referred to as the breakage frequency. When a particle breaks it disintegrates into a number of smaller fragments equal to $\gamma(\mathbf{x}, r, t)$; notice that this number, like the breakage frequency, depends on the position vector $\mathbf{x}$ and, more importantly, on the particle size r . If we know these functions, we can determine the total number of particles that breakage yields per unit time per unit volume of physical space $d\mathbf{x}$, which is given by:

$$\gamma(\mathbf{x}, r, t) b(\mathbf{x}, r, t) f_n(\mathbf{x}, r, t) dr \quad (3.4)$$

These fragments do not share the same size; hence, there exists a probability density function $p(s, |\mathbf{x}, r, t)$ that describes how the fragments are distributed over the state coordinate s . If we know this function we can write:

$$\gamma(\mathbf{x}, r, t)b(\mathbf{x}, r, t)f_n(\mathbf{x}, r, t)p(s, |\mathbf{x}, r, t)drds \quad (3.5)$$

to represent the number of particles with size range ds around s formed from the breakage of particles of size range dr around r per unit volume of physical space. If we only consider sizes greater than s breaking to give us fragments in the size range ds around s , then we can close the term $B_b(\mathbf{x}, s, t)$ thus:

$$\begin{aligned} B_b(\mathbf{x}, s, t)ds &= \int_s^{+\infty} \gamma(\mathbf{x}, r, t)b(\mathbf{x}, r, t)f_n(\mathbf{x}, r, t)p(s, |\mathbf{x}, r, t)drds \\ B_b(\mathbf{x}, s, t) &= \int_s^{+\infty} \gamma(\mathbf{x}, r, t)b(\mathbf{x}, r, t)f_n(\mathbf{x}, r, t)p(s, |\mathbf{x}, r, t)dr \end{aligned} \quad (3.6)$$

The breakage term $D_b(\mathbf{x}, s, t)$ defined above can also be closed thus:

$$b(\mathbf{x}, s, t)f_n(\mathbf{x}, s, t) \quad (3.7)$$

where $b(\mathbf{x}, s, t)f_n(\mathbf{x}, s, t)ds$ represents the average number of particles of size range ds around s breaking per unit time per unit volume of physical space. Putting the results in (3.6) and (3.7) together, we close the source term due to breakage $h_n^b(\mathbf{x}, s, t)$ thus :

$$h_n^b(\mathbf{x}, s, t) = \int_s^{+\infty} \gamma(\mathbf{x}, r, t)b(\mathbf{x}, r, t)f_n(\mathbf{x}, r, t)p(s, |\mathbf{x}, r, t)dr - b(\mathbf{x}, s, t)f_n(\mathbf{x}, s, t) \quad (3.8)$$

The expression is closed for it is written in terms of the density function and the particle state coordinates.

3.2.2 The Aggregation Equation

The generation term h_n in the PBE in part also accounts for aggregation. The first step in closing the aggregation term is to define the source and sink terms due to aggregation. Consequently we can write:

$$h_n^a(\mathbf{x}, s, t) = B_a(\mathbf{x}, s, t) - D_a(\mathbf{x}, s, t) \quad (3.9)$$

where

$B_a(\mathbf{x}, s, t)$ = the rate of birth of particles of size s due to aggregation; and

$D_a(\mathbf{x}, s, t)$ = the rate of death of particles of size s due to aggregation.

We summarise the assumptions made for the derivation of the aggregation equation as follows:

- 1) The possibility of having more than two particles simultaneously aggregating is extremely small and thus negligible;
- 2) No statistical correlation between the two aggregating particles is present.

Consider two particles with sizes dr around r and du around u colliding in such a way as to aggregate to yield another particle of size ds around s . If we define $a(\mathbf{x}, u, r, t)$ as the aggregation frequency which gives the fraction of ordered pairs aggregating per unit time, then the number of particles (of size in the range ds) that generate for any given pair of particles of sizes in the range du and dr in physical space is:

$$a(\mathbf{x}, u, r, t) f_n(\mathbf{x}, u, t) f_n(\mathbf{x}, r, t) du dr \quad (3.10)$$

We assume that given the particle state of one of the aggregating pair and that of the new particle, it is possible to solve for the particle state of the other aggregating pair. In this case, given the values of r and s , u is a function of the other two sizes: $u = u(s, r)$. For fixed r , we can write:

$$a[\mathbf{x}, u(s, r), r, t] f_n[\mathbf{x}, u(s, r), t] f_n(\mathbf{x}, r, t) \frac{du}{ds} ds dr \quad (3.11)$$

Only particles with complementary sizes below s can aggregate to yield a new particle of size s . Hence, we integrate over r for $0 < r < s$:

$$\left\{ \int_0^s a[\mathbf{x}, u(s, r), r, t] f_n[\mathbf{x}, u(s, r), t] f_n(\mathbf{x}, r, t) \frac{\partial u}{\partial s} dr \right\} ds \quad (3.12)$$

The expression (3.12) is equal to $B_a(\mathbf{x}, s, t) ds$. Hence,

$$B_a(\mathbf{x}, s, t) = \int_0^s a[\mathbf{x}, u(s, r), r, t] f_n[\mathbf{x}, u(s, r), t] f_n(\mathbf{x}, r, t) \frac{\partial u}{\partial s} dr \quad (3.13)$$

On the right hand side of (3.13), identical size pairs have been considered twice within the interval of integration so that multiplying by $\frac{1}{2}$ corrects for the redundancy. Therefore we write:

$$B_a(\mathbf{x}, s, t) = \frac{1}{2} \int_0^s a[\mathbf{x}, u(s, r), r, t] f_n[\mathbf{x}, u(s, r), t] f_n(\mathbf{x}, r, t) \frac{\partial u}{\partial s} dr \quad (3.14)$$

Similarly, we can close the term $D_a(\mathbf{x}, s, t)$ as follows:

$$\begin{aligned} D_a(\mathbf{x}, s, t) ds &= f_n(\mathbf{x}, s, t) \int_0^{+\infty} a(\mathbf{x}, s, r, t) f_n(\mathbf{x}, r, t) dr ds \\ D_a(\mathbf{x}, s, t) &= f_n(\mathbf{x}, s, t) \int_0^{+\infty} a(\mathbf{x}, s, r, t) f_n(\mathbf{x}, r, t) dr \end{aligned} \quad (3.15)$$

Hence we can close the source term due to aggregation $h_n^a(\mathbf{x}, s, t)$ thus:

$$\begin{aligned} h_n^a(\mathbf{x}, s, t) &= \frac{1}{2} \int_0^s a[\mathbf{x}, u(s, r), r, t] f_n[\mathbf{x}, u(s, r), t] f_n(\mathbf{x}, r, t) \frac{\partial u}{\partial s} dr - \\ &\quad f_n(\mathbf{x}, s, t) \int_0^{+\infty} a(\mathbf{x}, s, r, t) f_n(\mathbf{x}, r, t) dr \end{aligned} \quad (3.16)$$

3.2.3 The Population Balance Equation with Aggregation-Breakage Equation

For simplicity, let us consider a population of particles that can only aggregate. Consequently, the population balance equation includes only the accumulation and the aggregation terms. We assume that the state of the particles is characterized solely by the particle volume and that the system is uniform in real space. Consequently, the number density function $\hat{f}_n(v_p, t)$ depends only on the time t and on the particle volume v_p . In terms of this distribution function, the population balance equation reads :

$$\partial_t \hat{f}_n(v_p, t) = \hat{B}_a(v_p, t) - \hat{D}_a(v_p, t) \quad (3.17)$$

where:

$$\hat{B}_a(v_p, t) = \frac{1}{2} \int_0^{v_p} a(v_p - \varepsilon, \varepsilon) \hat{f}_n(v_p - \varepsilon, t) \hat{f}_n(\varepsilon, t) d\varepsilon \quad (3.18)$$

Note: $\frac{\partial(v_p - \varepsilon)}{\partial v_p}$ which is the volume equivalent of $\frac{\partial u}{\partial s}$ is equal to 1 in the above expression.

$$\hat{D}_a(v_p, t) = \hat{f}_n(v_p, t) \int_0^{+\infty} a(v_p, \varepsilon) \hat{f}_n(\varepsilon, t) d\varepsilon \quad (3.19)$$

$\hat{B}_a(v_p, t)$ = rate of birth of particles of volume v_p due to aggregation of smaller particles.
 $\hat{D}_a(v_p, t)$ = rate of death of particles of volume v_p due to aggregation with other particles.

Let us assume that we prefer to work in terms of particle size s instead of particle volume v_p . The NDF of interest is then a function of the time and the particle size; we denote it as $f_n(s, t)$. We now need to derive the PBE in terms of the size-based NDF. First, we let :

$$v_p(s) \equiv k_v s^3 \quad ; \quad \varepsilon(r) \equiv k_v r^3 \quad ; \quad v_p(s) - \varepsilon(r) \equiv k_v u^3 \quad (3.20)$$

where k_v represents the particle shape factor (assumed to be the same for all the particles of the population; this is consistent with the assumption that the particle size or volume is the only coordinate characterizing the particle state). Then, since it is :

$$f_n(s, t) ds = \hat{f}_n[v_p(s), t] dv_p(s) = \hat{f}_n[v_p(s), t] 3k_v s^2 ds \quad (3.21)$$

we obtain:

$$\hat{f}_n[v_p(s), t] = \frac{f_n(s, t)}{3k_v s^2} \quad ; \quad \hat{f}_n[\varepsilon(r), t] = \frac{f_n(r, t)}{3k_v r^2} \quad ; \quad \hat{f}_n[v_p(s) - \varepsilon(r), t] = \frac{f_n(u, t)}{3k_v u^2} \quad (3.22)$$

Hence we can write:

$$\begin{aligned} \hat{B}_a[v_p(s), t] &= \frac{1}{2} \int_0^s \{a[v_p(s) - \varepsilon(r), \varepsilon(r)] / 3k_v u^2\} f_n(u, t) f_n(r, t) dr \\ \hat{D}_a[v_p(s), t] &= \frac{f_n(s, t)}{3k_v s^2} \int_0^{+\infty} a[v_p(s), \varepsilon(r)] f_n(r, t) dr \end{aligned} \quad (3.23)$$

Accordingly, the original PBE:

$$\partial_t \hat{f}_n(v_p(s), t) = \hat{B}_a(v_p(s), t) - \hat{D}_a(v_p(s), t) \quad (3.24)$$

results to:

$$\partial_t f_n(s, t) = 3k_v s^2 \hat{B}_a[v_p(s), t] - 3k_v s^2 \hat{D}_a[v_p(s), t] = B_a(s, t) - D_a(s, t) \quad (3.25)$$

where:

$$B_a(s, t) = \frac{1}{2} s^2 \int_0^s \{a[v_p(s) - \varepsilon(r), \varepsilon(r)]/u^2\} f_n(u, t) f_n(r, t) dr \quad (3.26)$$

$$D_a(s, t) = f_n(s, t) \int_0^{+\infty} a[v_p(s), \varepsilon(r)] f_n(r, t) dr \quad (3.27)$$

Similarly, if we consider a population of particles undergoing breakage only. Consequently, the population balance equation includes only the accumulation and the breakage terms. We assume that the state of the particles is characterized solely by the particle volume and that the system is uniform in real space. Consequently, the number density function $\hat{f}_n(v_p, t)$ depends only on the time t and on the particle volume v_p . In terms of this distribution function, the population balance equation reads :

$$\partial_t \hat{f}_n(v_p, t) = \hat{B}_b(v_p, t) - \hat{D}_b(v_p, t) \quad (3.28)$$

where:

$$\hat{B}_b(v_p, t) = \int_{v_p}^{+\infty} \hat{b}(\varepsilon, t) \hat{f}_n(\varepsilon, t) \hat{P}(v_p | \varepsilon, t) d\varepsilon \quad (3.29)$$

$$\hat{D}_b(v_p, t) = \hat{b}(v_p, t) \hat{f}_n(v_p, t) \quad (3.30)$$

$\hat{B}_b(v_p, t)$ = rate of birth of particles of volume v_p due to fragmentation of bigger particles.
 $\hat{D}_b(v_p, t)$ = rate of death of particles of volume v_p due to fragmentation.

In equation (3.26), $\hat{P}(v_p | \varepsilon, t)$ contains information on the fragments produced by a breakage event. This is the daughter distribution function. Where $3k_v s^2 \hat{P}(v_p | \varepsilon, t) = P(s | r, t)$ and $\hat{b}(v_p, t) = b(s, t)$, we can re-write equations (3.29) and (3.30) as follows:

$$\begin{aligned} \frac{B_b(s, t)}{3k_v s^2} &= \int_s^{+\infty} b(r, t) \frac{f_n(r, t)}{3k_v r^2} \frac{P(s | r, t)}{3k_v s^2} 3k_v r^2 dr \\ B_b(s, t) &= \int_s^{+\infty} b(r, t) f_n(r, t) P(s | r, t) dr \end{aligned} \quad (3.31)$$

$$\begin{aligned} \frac{D_b(s, t)}{3k_v s^2} &= b(s, t) \frac{f_n(s, t)}{3k_v s^2} \\ D_b(s, t) &= b(s, t) f_n(s, t) \end{aligned} \quad (3.32)$$

Hence the PBE with aggregation and breakage equations is written as:

$$\partial_t f_n + \nabla_x \cdot (f_n \langle v | s \rangle) + \partial_s (f_n s | s) = B_a(s, t) - D_a(s, t) + B_b(s, t) - D_b(s, t) \quad (3.33)$$

after substituting (3.26), (3.27), (3.31) and (3.32), equation (3.33) becomes:

$$\begin{aligned} \partial_t f_n + \nabla_x \cdot (f_n \langle \mathbf{v} | s \rangle) + \partial_s (f_n s | s) &= \frac{s^2}{2} \int_0^s \{a[v_p(s) - \varepsilon(r), \varepsilon(r)]/u^2\} f_n(u, t) f_n(r, t) dr \\ &- f_n(s, t) \int_0^\infty a[v_p(s), \varepsilon(r)] f_n(r, t) dr + \int_s^\infty b(r, t) f_n(r, t) P(s|r, t) dr - b(s, t) f_n(s, t) \end{aligned} \quad (3.34)$$

While trying to find a solution for the left hand side of equation (3.34), assuming h_n is closed, we applied the DQMOM where the quadrature approximation was employed and the moment transform of the resulting expression was found. The next section shall consider the moment of the generation term h_n .

3.2.4 Moment Equation of the Generation term h_n

This section resumes from equation (2.63). We recall this equation here:

$$(1 - k) \sum_{i=1}^v c_i^n s_i^k + k \sum_{i=1}^v c_i^s s_i^{k-1} = \mathcal{M}_k(h_n) + k \sum_{i=1}^v n_i s_i^{k-1} s_i$$

h_n is the generation term. We can now evaluate this expression knowing that h_n can be written in terms of the NDF. Recall:

$$h_n(s) = B_a(s) - D_a(s) + B_b(s) - D_b(s) \quad (3.35)$$

Let us consider the first term on the right hand side of equation (3.35):

$$B_a(s) = (s^2/2) \int_0^s \frac{\{a[u(s, r), r]\}}{u^2(s, r)} f_n[u(s, r)] f_n(r) dr \quad (3.36)$$

with:

$$u(s, r) = (s^3 - r^3)^{1/3} \quad (3.37)$$

The function $a(u, r)$ denotes the fraction of particles of sizes u and r that aggregate per unit time. Of course, this function is zero for negative values of the particle size; in particular, $a(u, r) \equiv 0$ for $u < 0$. Accordingly, we can write Eq. (3.36) equivalently as:

$$B_a(s) = (s^2/2) \int_0^{+\infty} \frac{\{a[u(s, r), r]\}}{u^2(s, r)} f_n[u(s, r)] f_n(r) dr \quad (3.38)$$

This is because for $r > s$ the value of u is negative, and therefore $a(u, r)$ is zero. In other words, the added part of the integral (with integration limits between s and infinity) is zero; so, including it does not alter the final result. We just have to ensure that $a(u, r)$ is properly defined.

Let us consider the moment of order k of the function $B_a(s)$. To calculate it we integrate the function $s^k B_a(s)$ over s between zero and infinity. Thus, it is:

$$\mathcal{M}_k\{B_a(s)\} = (1/2) \int_0^{+\infty} s^{2+k} \left[\int_0^{+\infty} \frac{\{a[u(s, r), r]\}}{u^2(s, r)} f_n[u(s, r)] f_n(r) dr \right] ds \quad (3.39)$$

Here, we first calculate the inner integral (the one between brackets; doing so yields a function of s) and then the outer integral. We find it convenient, however, to change the integration variable s into u . To this end, from Eq. (3.37), we see that $ds = u^2/(u^3 + r^3)^{2/3} du$. In this expression, we regard r as fixed. When s tends to infinity, u tends to infinity; when s tends to zero, u tends to a negative value, and therefore - since for $u < 0$ the function $a(u, r)$ is zero by definition - the lower integration limit can be set to zero. So, after changing the integration variable, the integral above becomes:

$$\mathcal{M}_k\{B_a(s)\} = (1/2) \int_0^{+\infty} f_n(r) \int_0^{+\infty} s^k(u, r) a(u, r) f_n(u) du dr \quad (3.40)$$

To calculate the integrals, we write:

$$f_n(u) = \sum_{i=1}^v n_i \delta(u - s_i) \quad ; \quad f_n(r) = \sum_{j=1}^v n_j \delta(r - s_j) \quad (3.41)$$

Thus, it is :

$$\begin{aligned} \int_0^{+\infty} s^k(u, r) a(u, r) f_n(u) du &= \sum_{i=1}^v n_i \int_0^{+\infty} s^k(u, r) a(u, r) \delta(u - s_i) du \\ &= \sum_{i=1}^v n_i s^k(s_i, r) a(s_i, r) \end{aligned} \quad (3.42)$$

Similarly, we write:

$$\begin{aligned} \mathcal{M}_k\{B_a(s)\} &= (1/2) \sum_{i=1}^v \sum_{j=1}^v n_i n_j \int_0^{+\infty} s^k(s_i, r) a(s_i, r) \delta(r - s_j) dr \\ &= (1/2) \sum_{i=1}^v \sum_{j=1}^v n_i n_j s^k(s_i, s_j) a(s_i, s_j) \end{aligned} \quad (3.43)$$

Let us consider the second term on the right hand side of Eq. (3.35)

$$D_a(s) = f_n(s) \int_0^{+\infty} a(s, r) f_n(r) dr \quad (3.44)$$

The function $a(s, r)$ denotes the fraction of particles of sizes s and r that aggregate per unit time. Let us consider the moment of order k of the function $D_a(s)$. To calculate it we integrate the function $s^k D_a(s)$ over s between zero and infinity. Thus it is:

$$\mathcal{M}_k\{D_a(s)\} = \int_0^{+\infty} s^k(s) f_n(s) \left[\int_0^{+\infty} a(s, r) f_n(r) dr \right] ds \quad (3.45)$$

Here, we first calculate the inner integral (the one between the brackets; doing so yields a function of s) and then the outer integral. To calculate the integrals, we write:

$$f_n(s) = \sum_{i=1}^v n_i \delta(s - s_i) \quad ; \quad f_n(r) = \sum_{j=1}^v n_j \delta(r - s_j) \quad (3.46)$$

Thus it is:

$$\begin{aligned}\int_0^{+\infty} a(s, r) f_n(r) dr &= \sum_{j=1}^v n_j \int_0^{+\infty} a(s, r) \delta(r - s_j) dr \\ &= \sum_{j=1}^v n_j a(s, s_j)\end{aligned}\quad (3.47)$$

Similarly, we write:

$$\begin{aligned}\mathcal{M}_k\{D_a(s)\} &= \sum_{i=1}^v \sum_{j=1}^v n_i n_j \int_0^{+\infty} s^k(s) a(s, s_j) \delta(s - s_i) ds \\ &= \sum_{i=1}^v \sum_{j=1}^v n_i n_j s^k(s_i) a(s_i, s_j)\end{aligned}\quad (3.48)$$

Let us consider the third term on the right hand side of Eq. (3.35)

$$B_b(s) = \int_s^{+\infty} b(r) f_n(r) P(s|r) dr \quad (3.49)$$

Here, the lower limit of integration can be set to zero. This is because the function $P(s|r)$ is zero for values of $s < 0$. Particles with sizes less than s can only break to form particles with sizes less than s . Hence, the probability of $P(s|r)$ automatically equal zero. We can therefore write:

$$B_b(s) = \int_0^{+\infty} b(r) f_n(r) P(s|r) dr \quad (3.50)$$

Let us consider the moment of order k of the function $B_b(s)$. To calculate it, we integrate the function $s^k B_b(s)$ over s between zero and infinity. Thus it is:

$$\mathcal{M}_k\{B_b(s)\} = \int_0^{+\infty} s^k(s) \left[\int_0^{+\infty} b(r) f_n(r) P(s|r) dr \right] ds \quad (3.51)$$

To calculate the integrals, we write:

$$f_n(r) = \sum_{i=1}^v n_i \delta(r - s_i) \quad (3.52)$$

We first calculate the inner integral.

$$\begin{aligned}\int_0^{+\infty} b(r) f_n(r) P(s|r) dr &= \sum_{i=1}^v n_i \int_0^{+\infty} b(r) P(s|r) \delta(r - s_i) dr \\ &= \sum_{i=1}^v n_i b(s_i) P(s|s_i)\end{aligned}\quad (3.53)$$

and then the outer integral whose integrand is now a function of s

$$\begin{aligned}\mathcal{M}_k\{B_b(s)\} &= \sum_{i=1}^v n_i b(s_i) \int_0^{+\infty} s^k P(s|s_i) ds \\ &= \sum_{i=1}^v n_i b(s_i) P_i^{(k)}\end{aligned}\quad (3.54)$$

where

$$P_i^{(k)} = \int_0^{+\infty} s^k P(s|s_i) ds \quad (3.55)$$

The fourth term of Eq. (3.35) is given as:

$$D_b(s) = b(s)f_n(s) \quad (3.56)$$

Let us consider the moment of order k of the function $D_b(s)$. To calculate it, we integrate the function $s^k D_b(s)$ over s between zero and infinity. Thus it is:

$$\mathcal{M}_k\{D_b(s)\} = \int_0^{+\infty} s^k(s)b(s)f_n(s)ds \quad (3.57)$$

To calculate the integral, we write:

$$f_n(s) = \sum_{i=1}^v n_i \delta(s - s_i) \quad (3.58)$$

Therefore we write:

$$\begin{aligned} \mathcal{M}_k\{D_b(s)\} &= \sum_{i=1}^v n_i \int_0^{+\infty} s^k(s)b(s)\delta(s - s_i)ds \\ &= \sum_{i=1}^v n_i s^k(s_i)b(s_i) \end{aligned} \quad (3.59)$$

Therefore we can write an expression for $\mathcal{M}_k(h_n)$ in equation (2.61) as follows:

$$\begin{aligned} \mathcal{M}_k(h_n) &= (1/2) \sum_{i=1}^v \sum_{j=1}^v n_i n_j s^k(s_i, s_j) a(s_i, s_j) - \sum_{i=1}^v \sum_{j=1}^v n_i n_j s^k(s_i) a(s_i, s_j) + \sum_{i=1}^v n_i b(s_i) P_i^{(k)} - \sum_{i=1}^v n_i s^k(s_i) b(s_i) \\ &= (1/2) \sum_{i=1}^v \sum_{j=1}^v n_i n_j a(s_i, s_j) \left[s^k(s_i, s_j) - 2s^k(s_i) \right] + \sum_{i=1}^v n_i b(s_i) \left[P_i^{(k)} - s^k(s_i) \right] \end{aligned} \quad (3.60)$$

3.3 Modelling the Aggregation and Breakage Kernel

In the outgoing section we were able to write a full PBE including the aggregation and breakage equations. In equation (3.60), we encountered the aggregation and breakage kernels which are effective parameters in the PBE. These parameters are mainly made of two parts:

- 1) a part which informs us of the number of particle pairs available for collision or the number of collisions undergone by each particle per time. This is the *collision frequency*.
- 2) a second part which gives information on the success of aggregation or breakage resulting from these collisions. This is the *success factor for aggregation or breakage*. We also refer to this as the *aggregation or breakage efficiency* in this work.

Hence the aggregation or breakage kernel takes the generic form:

$$k(s) = \Psi_0 k^*(s) \quad (3.61)$$

where Ψ_0 is the success factor for aggregation or breakage which is dependent on the operating conditions and various process parameters like kinetic energy trajectories, coalescence mechanism, binder properties, particle characteristics, collision orientation etc. $k^*(s)$ is the collision frequency which is size-dependent. Developing apriori expressions from a theoretical standpoint alone to represent $k(s)$ is a difficult task. The ultimate functional form of $k(s)$ must be determined with the aid of experimental data. Different forms of the collision frequency $k^*(s)$ exists in the literature. In this work, we adopt the kinetic theory of granular flow (KTGF) collision model. Previous work conducted by Goldschmidt (2001) shows that we can confidently rely on the KTGF approximation to describe the behaviour of particle motion in a fluidized bed. The reader is referred to the work of Goldschmidt (2001) for more details.

3.3.1 The Aggregation Kernel

From the work of Goldschmidt (2001) we can derive an aggregation kernel from the KTGF expression which describes the collision rate for particles in a multi-component mixture. The number of collisions per unit physical volume and time between particles of sizes s_i and s_j can be written as (Goldschmidt, 2001; Fan et al., 2004) :

$$N_{ij} = \pi n_i n_j \sigma_{ij}^3 g_{ij} \left[\frac{4}{\sigma_{ij}} \left(\frac{\theta_s m_i + m_j}{\pi 2 m_i m_j} \right)^{\frac{1}{2}} - \frac{2}{3} (\nabla \cdot \bar{v}) \right] \quad (3.62)$$

where σ_{ij} is the inter-particle distance between the two colliding particles of sizes s_i and s_j , g_{ij} is the radial distribution for the mixture, θ_s is the mixture granular temperature, m_i and m_j are the masses of particle sizes s_i and s_j respectively and $\bar{v}$ is the ensemble average particle velocity. From the law of conservation of linear momentum, $\bar{v} = \phi_i v_i + \phi_j v_j$, where ϕ_i , v_i and ϕ_j , v_j are the volume fractions and velocities of particles i and j respectively.

If we compare Eq.(3.43) to Eq.(3.62), we can say that the product $n_i n_j$ gives information on the total number of ordered pairs of particles with sizes s_i and s_j available for collision/aggregation per unit physical space volume. Thus, the aggregation kernel can be expressed as:

$$\Psi_a \pi \sigma_{\alpha\beta}^3 g_{\alpha\beta} \left[\frac{4}{\sigma_{\alpha\beta}} \left(\frac{\theta_s m_\alpha + m_\beta}{\pi 2 m_\alpha m_\beta} \right)^{\frac{1}{2}} - \frac{2}{3} (\nabla \cdot v) \right] \quad (3.63)$$

where the expression

$$\pi \sigma_{\alpha\beta}^3 g_{\alpha\beta} \left[\frac{4}{\sigma_{\alpha\beta}} \left(\frac{\theta_s m_\alpha + m_\beta}{\pi 2 m_\alpha m_\beta} \right)^{\frac{1}{2}} - \frac{2}{3} (\nabla \cdot v) \right]$$

is the collision frequency which gives the number of collisions per unit time between particles of sizes s_i and s_j . Ψ_a is the success-factor for aggregation which we shall refer to

Table 3.1: Constitutive relations for the solids collision parameters

<i>Solids Collision Parameters</i>
$g_{ij} = \frac{s_i g_j + s_j g_i}{s_i + s_j}$
$\sigma_{ij} = (s_i + s_j)/2$
$\theta_s = \frac{\phi_i \rho_i \theta_i + \phi_j \rho_j \theta_j}{\phi_i \rho_i + \phi_j \rho_j} (m_i + m_j)$
$m_i = \rho_i \pi s_i^3 / 6$

as the *aggregation efficiency* in this work and it is dependent on the operating conditions and material properties that contribute to particle aggregation such as the binder viscosity, particle size, particle velocity, fluidizing velocity and bed temperature.

If we neglect the divergence of the particle velocity field and assume that particles have same density while substituting for the inter-particle distance, σ_{ij} equation (3.63) reduces to:

$$a(s_i, s_j) = \Psi_a g_{ij} \left(\frac{3\theta_s}{\rho_s} \right)^{\frac{1}{2}} (s_i + s_j)^2 \left(\frac{1}{s_i^3} + \frac{1}{s_j^3} \right)^{\frac{1}{2}} \quad (3.64)$$

The equation (3.64) represents the kinetic aggregation kernel. Refer to Table 3.1 for a summary of the solids collision parameters. Other associated parameters, for example, the granular temperature of the solid (θ_i), are defined in section 4.1.

3.3.2 The Breakage Kernel

The breakage kernel $b(s_i)$ can be written in the form of Eq. (3.61) as a product of the size-dependent collision frequency $b^*(s_i)$ and a breakage efficiency Ψ_b . Thus

$$b(s_i) = \Psi_b b^*(s_i) \quad (3.65)$$

Here, we have assumed fragmentation to be a direct result of collision. From the kinetic theory of granular flow, the average number of collisions undergone by each particle per time is called the collision frequency. The frequency of collisions for a particle of size s_i with particles of different sizes is given by:

$$\frac{N_{i1} + N_{i2} + N_{i3} + \dots}{n_i} = \frac{1}{n_i} \sum_{j=1}^v N_{ij} \quad (3.66)$$

Substituting equation (3.62) into (3.66) we have :

$$b^*(s_i) = \frac{1}{n_i} \sum_{j=1}^v \pi n_i n_j \sigma_{ij}^3 g_{ij} \left[\frac{4}{\sigma_{ij}} \left(\frac{\theta_s}{\pi} \frac{m_i + m_j}{2m_i m_j} \right)^{\frac{1}{2}} - \frac{2}{3} (\nabla \cdot \vec{v}) \right] \quad (3.67)$$

which reduces to:

$$b^*(s_i) = \sum_{j=1}^v \pi n_j \sigma_{ij}^3 g_{ij} \left[\frac{4}{\sigma_{ij}} \left(\frac{\theta_s m_i + m_j}{\pi 2m_i m_j} \right)^{\frac{1}{2}} - \frac{2}{3} (\nabla \cdot \bar{\mathbf{v}}) \right] \quad (3.68)$$

Therefore the breakage kernel can be expressed as:

$$b(s_i) = \Psi_b \sum_{j=1}^v n_j \pi \sigma_{ij}^3 g_{ij} \left[\frac{4}{\sigma_{ij}} \left(\frac{\theta_s m_i + m_j}{\pi 2m_i m_j} \right)^{\frac{1}{2}} - \frac{2}{3} (\nabla \cdot \bar{\mathbf{v}}) \right] \quad (3.69)$$

where Ψ_b is the success factor for breakage or the breakage efficiency.

If we neglect the divergence of the particle velocity field and assume that particles have equal density, we can write the breakage kernel as :

$$b(s_i) = \Psi_b \sum_{j=1}^v n_j g_{ij} \left(\frac{3\theta_s}{\rho_s} \right)^{\frac{1}{2}} (s_i + s_j)^2 \left(\frac{1}{s_i^3} + \frac{1}{s_j^3} \right)^{\frac{1}{2}} \quad (3.70)$$

3.3.3 The Daughter Distribution Function

While deriving the breakage equation in § 3.2.1, we introduced the probability density function (PDF) $p(s, |\mathbf{x}, r, t)$ to quantify the rate of formation of particles due to the first-order breakage process; First-order because only one particle takes place in the discontinuous event and so far in this work the breakage rate has been assumed to be proportional to the NDF. The conditional PDF $p(s, |\mathbf{x}, r, t)$ states the probability of forming a daughter particle of size s from a parent particle of size r . However, this PDF inherits certain properties from the conservation law which helps to guide the breakage process. One of these constraints is the normalization condition given as:

$$\int_{\Omega_s} p(s, |\mathbf{x}, r, t) ds = 1 \quad (3.71)$$

which states that the probability of forming the daughter particles is unity.

In most cases a closely similar conditional PDF is used instead. This corresponding conditional PDF was introduced in § 3.2.4. It is the *daughter distribution function* which in addition to giving the probability of forming a daughter particle of a given size from a particular parent particle size, also contains information on the number of daughter particles formed from a breakage event. Let $P(s|\mathbf{x}, r, t)$ be the daughter distribution function, then a simple renormalization of the PDF constraint will yield:

$$\int_{\Omega_s} P(s|r) ds = \gamma(r) \quad (3.72)$$

where $\gamma(r)$ is the number of new particles formed from the breakage event (Note: here we have suppressed the $\mathbf{x}$ and t dependencies for clarity).

Different scenarios are possible in considering a suitable daughter distribution function. The simplest functional form being a summation of dirac deltas:

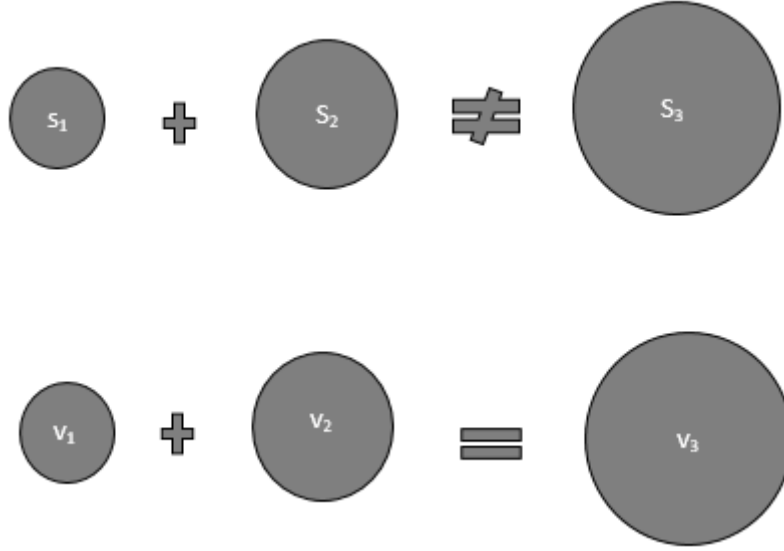


Figure 3.1: The additive nature of volume-based vs. size-based intrinsic particle property

$$P(s|r) = \sum_{i=1}^{\gamma} \delta[s - s_i(r)] \quad (3.73)$$

where γ is the total number of daughter particles and s_i relates the size of the i th daughter particle to that of the parent r . In other words Eq. (3.73) states that when a particle breaks, γ fragments are formed and s_i is the size characterizing each fragment. With respect to the size of the fragments and the parent particle some additional constraints have to be introduced. In order to introduce these constraints we first introduce volume-based sizes as the intrinsic particle property. This is because particle volume is additive whereas particle size (for example diameter) is not. That is $v_3 = v_1 + v_2$ and $s_3 \neq s_1 + s_2$ where the subscripts 1 and 2 refer to the daughter particles and 3 to the parent particle. See schematic representation in Figure 3.1. We can therefore write:

$$\int_{\Omega_\varepsilon} v_p(s) \hat{P}[v_p(s)|\varepsilon(r)] dv_p \leq \varepsilon(r) \quad (3.74)$$

The equality sign holds if there were no loss of mass (or volume) during breakage. In this work, we shall assume this to be the case.

If the j th moment of the daughter distribution ($\hat{P}_j$) is written as a function of the volume of the particle. Then, we can write:

$$\hat{P}_j = \int_{\Omega_\varepsilon} v_p^j(s) \hat{P}[v_p(s)|\varepsilon(r)] dv_p \quad (3.75)$$

and the number and mass balance imposed constraints in Eqs. (3.72) and (3.74) will represent the zeroth and first moment respectively. To impose the positiveness of the distribution and conservation of mass, we introduce one more constraint as follows (Marchisio and Fox, 2013):

$$\hat{P}[v_p(s)|\varepsilon(r)] = \begin{cases} 0 & \text{if } v_p < 0 \\ 0 & \text{if } v_p > \varepsilon \end{cases}$$

where the probability of forming fragments with volume greater than that of the parent or less than zero is null. If we substitute the relation $v_p = k_v s^3$ and $dv_p = 3k_v s^2 ds$ into equations (3.74) and (3.75) we get their equivalent expressions in terms of particle sizes.

In the literature, there are many daughter distribution functions. In principle, since we do not know apriori which daughter distribution function best describes our system, we will need to consider a number of daughter distribution functions from literature and test them with our model and compare the results with experimental data to ascertain which gives an accurate description of our system. However, at the moment, we shall employ a closure which makes the problem more tractable. We shall employ a daughter distribution function with binary breakage. There are two possibilities: The first is a symmetric breakage where the two fragments are identical. Here, $m = n$ where m and n are the mass ratios between the two daughter particles formed (for example, if $m = 1$ and $n = 1$, the two fragments have the same volume and symmetric fragmentation can be considered). The second is asymmetric breakage where $m \gg n$ or vice-versa. As regards the daughter distribution function, the following relation will be used (Marchisio et al., 2003b):

$$P_i^{(k)} = \int_0^{+\infty} s^k P(s|r) ds = r^k \frac{m^{k/3} + n^{k/3}}{(m+n)^{k/3}} \quad (3.76)$$

In this work, we shall consider the symmetric breakage in which $m = 1$ and $n = 1$.

Chapter 4

Model Description and Methodology

4.1 Mathematical Models

Modelling the behaviour of dense fluidized bed suspensions is a very complex task involving a lot of related phenomena and interactions such as mass transfer, heat transfer, chemical reactions, growth, breakage, aggregation and so on. Numerical solutions for the continuity, momentum, energy and chemical species equations will therefore be needed (Fan et al., 2004). As earlier reported in § 2.2, the disperse phase also needs to be modelled using the PBE. To render our work less complex we shall assume isothermal condition for the fluidized bed with no chemical reactions. The PSD changes mainly due to aggregation and breakage and no growth in size space. Fan et al. (2004) used the MFIx (Multiphase Flow with Interphase eXchanges) open source CFD code and models/constitutive equations customary to the software in their work. Some of these constitutive relationships are however not available in Fluent. Therefore, we shall select closely related formulations in Fluent and also implement some of the models using User Defined Functions (UDFs).

4.1.1 The Multi-Fluid Model for Gas-Solid Flow

The gas and solid phases will be modelled using the Eulerian modelling approach where both the primary phase (gas) and the secondary or dispersed phases (solids) are treated as inter-penetrating continua. Each solid phase has its discrete properties e.g. size. The volume fractions of both phases (primary and secondary) sums up to unity.

$$\phi_g + \sum_{i=1}^v \phi_{si} = 1 \quad (4.1)$$

where ϕ_g is the gas volume fraction, ϕ_{si} is the volume fraction of the i^{th} solid phase, and v is the total number of solid phases. The continuity equations for the gas and i^{th} solid phase are:

$$\partial_t(\phi_g \rho_g) + \nabla \cdot (\phi_g \rho_g \mathbf{u}_g) = 0 \quad (4.2)$$

$$\partial_t(\phi_{si} \rho_{si}) + \nabla \cdot (\phi_{si} \rho_{si} \mathbf{u}_{si}) = 0 \quad (4.3)$$

where ρ_{si} , ρ_g and $\mathbf{u}_{si}$, $\mathbf{u}_g$ are the densities of the i^{th} solid phase and gas phase and the velocities of the i^{th} solid phase and the gas phase respectively. The right-hand side of

equation (4.3) is not equal to zero when aggregation and breakage are considered. The momentum balance for the gas phase is

$$\partial_t(\phi_g \rho_g \mathbf{u}_g) + \nabla \cdot (\phi_g \rho_g \mathbf{u}_g \mathbf{u}_g) = \nabla \cdot \mathbf{S}_g + \sum_{i=1}^v \mathbf{f}_{gi} + \phi_g \rho_g \mathbf{g} \quad (4.4)$$

Similarly, for the solid phases, the momentum balances are

$$\partial_t(\phi_{si} \rho_{si} \mathbf{u}_{si}) + \nabla \cdot (\phi_{si} \rho_{si} \mathbf{u}_{si} \mathbf{u}_{si}) = \nabla \cdot \mathbf{S}_{si} - \mathbf{f}_{gi} + \sum_{j=1}^v \mathbf{f}_{ij} + \phi_{si} \rho_{si} \mathbf{g} \quad (4.5)$$

where $\mathbf{S}_g$ and $\mathbf{S}_{si}$ are the stress tensors for the gas and the solid phases respectively, $\mathbf{g}$ is gravitational acceleration, $\mathbf{f}_{gi}$ is the interaction force between the gas phase and the i_{th} solid phase whereas the interaction force between the i_{th} and j_{th} solid phases is denoted by $\mathbf{f}_{ij}$.

The gas-solid interaction forces consists of buoyancy, drag, lift, Faxen, virtual mass and an history-dependent force similar to the Basset force for the motion of single particles. In this work only the drag and buoyancy forces are accounted for. The additional contributions are neglected for reasons already discussed in Owoyemi et al. (2007). We close the drag correlation for the fluid-particle interaction using the expression of Gidaspow (1994):

$$\mathbf{f}_{gi} = -\phi_{si} \nabla P_g - F_{gi}(\mathbf{u}_{si} - \mathbf{u}) \quad (4.6)$$

where

$$F_{gi} = \begin{cases} 150 \frac{\phi_{si}^2 \mu_g}{\phi_g s_i^2} + 1.75 \frac{\phi_{si} \rho_g |\mathbf{u}_g - \mathbf{u}_{si}|}{s_i} & \text{if } \phi_g \leq 0.8 \\ \frac{3}{4} C_D \frac{\phi_{si} \rho_g \phi_g |\mathbf{u}_g - \mathbf{u}_{si}|}{s_i} \phi_g^{-2.65} & \text{if } \phi_g > 0.8 \end{cases} \quad (4.7)$$

and

$$C_D = \begin{cases} \frac{24}{Re_{si}} (1 + 0.15 Re_{si}^{0.687}) & \text{if } Re_{si} \leq 1000 \\ 0.44 & \text{if } Re_{si} > 1000 \end{cases} \quad (4.8)$$

where Re_{si} and C_D are the Reynolds number and drag co-efficient respectively. The Particle Reynold's number is calculated thus:

$$Re_{si} = \frac{\phi_g \rho_g |\mathbf{u}_g - \mathbf{u}_{si}| s_i}{\mu_g} \quad (4.9)$$

We have also assumed that the interaction force between particles of the different solid phases include only the drag contribution described by Syamlal et al. (1993):

$$\mathbf{f}_{ij} = -F_{ij}(\mathbf{u}_{sj} - \mathbf{u}_{si}) \quad (4.10)$$

where

$$F_{ij} = \frac{3(1 + e_{ij})(\frac{\pi}{2} + \frac{C_{fij}\pi^2}{8})\phi_{si}\rho_{si}\phi_{sj}\rho_{sj}(s_i + s_j)^2 g_{ij} |\mathbf{u}_{si} - \mathbf{u}_{sj}|}{2\pi(\rho_{si}s_i^3 + \rho_{sj}s_j^3)} \quad (4.11)$$

e_{ij} is the co-efficient of restitution set at 0.8, C_{fij} is the co-efficient of friction equal to 0.15 and g_{ij} is the radial distribution obtained by combining the radial distribution functions g_i and g_j of the i -th and j -th solid particles phases respectively. The radial distribution employed in MFIx and used in Fan et al. (2004) is derived by Lebowitz (1964) and expressed as:

$$g_i = \frac{1}{\phi_g} + \frac{3s_i}{2\phi_g^2} \sum_{\lambda=1}^v \frac{\phi_{s\lambda}}{s_\lambda} \quad ; \quad g_{ij} = \frac{1}{\phi_g} + \frac{3s_i s_j}{\phi_g^2 (s_i + s_j)} \sum_{\lambda=1}^v \frac{\phi_{s\lambda}}{s_\lambda} \quad (4.12)$$

for each solid phase. In the literature, there is no unique formulation for the radial distribution and in ANSYS Fluent a number of options are available. For this work, we use the expression derived by Iddir and Arastoopour (2005). Equation 5.7 is a generalization of the Percus-Yevick equation for a mixture of hard spheres which takes into account a direct and indirect influence of particle i on particle j and all other particles respectively. At high solid volume fraction (when $\phi_{s\lambda}$ approaches $\phi_{s,max}$ the maximum packing limit), g_i diverges. Eq. 5.7 was modified by Iddir and Arastoopour (2005) to correctly mimic the results from Alder and Wainwright molecular dynamic simulations as follows:

$$g_i = [1 - (\frac{\phi_s}{\phi_{s,max}})]^{-1} + \frac{3s_i}{2} \sum_{\lambda=1}^N \frac{\phi_{s\lambda}}{s_\lambda} \quad (4.13)$$

where N is the total number of solid phases, $\phi_{s,max}$ is the packing limit and

$$\phi_s = \sum_{\lambda=1}^N \phi_{s\lambda}$$

λ refers to the solid phases only and g_{ij} is evaluated as reported in Table 3.1.

The effective stress tensors for the gas-phase and solid phase is closed using customary Newtonian constitutive equations (Jackson, 2000) as follows:

$$\mathbf{S}_g = -P_g \mathbf{I} + \boldsymbol{\tau}_g \quad ; \quad \boldsymbol{\tau}_g = 2\kappa_g \mu_g \mathbf{D}_g - 2/3 \kappa_g \mu_g Tr(\mathbf{D}_g) \mathbf{I} \quad ; \quad \mathbf{D}_g = 1/2 [\nabla \mathbf{u}_g + (\nabla \mathbf{u}_g)^T] \quad (4.14)$$

$$\mathbf{S}_{si} = -P_{si} \mathbf{I} + \boldsymbol{\tau}_{si} \quad ; \quad \boldsymbol{\tau}_{si} = 2\kappa_{si} \mu_{si} \mathbf{D}_{si} - 2/3 \kappa_{si} \mu_{si} Tr(\mathbf{D}_{si}) \mathbf{I} \quad ; \quad \mathbf{D}_{si} = 1/2 [\nabla \mathbf{u}_{si} + (\nabla \mathbf{u}_{si})^T] \quad (4.15)$$

where P_g , P_{si} , κ_g , κ_{si} , μ_g , μ_{si} , $\mathbf{D}_g$, $\mathbf{D}_{si}$, are the average pressures, the dilatational viscosities, the shear viscosities and the rate of deformation (or strain) tensor of the fluid and solid phases respectively. $\mathbf{I}$ is the identity tensor. The strain tensor is defined as:

$$\mathbf{D}_g = 1/2 [\nabla \mathbf{u}_g + (\nabla \mathbf{u}_g)^T] \quad ; \quad \mathbf{D}_{si} = 1/2 [\nabla \mathbf{u}_{si} + (\nabla \mathbf{u}_{si})^T] \quad (4.16)$$

Basically, closing the stress tensors will mean finding appropriate closures for the pressure, shear viscosities and the dilatational viscosities. The fluid is modelled using the ideal gas law and is treated as incompressible. Since the fluid density is constant and the pressure equation greatly simplified, we will not specify the fluid pressure constitutively. In addition, we assume that μ_g is constant and κ_g negligible. However, for the solid phase we shall employ two different methods to calculate the solid stress tensor in two differing regimes: the viscous and the plastic regimes. Theories from soil mechanics are used for the plastic or slowly shearing regime whereas for the viscous regime the kinetic theory is used.

In both regimes, each granular phase is usually characterized by a viscous solid pressure P_{si}^v , a plastic solid pressure, P_{si}^p , a viscous shear viscosity μ_{si}^v , a plastic shear viscosity μ_{si}^p , a viscous dilatational viscosity κ_{si}^v and a plastic dilatational viscosity κ_{si}^p . Generally, if in the viscous regime a solid property x_{si} exists and equals x_{si}^v , it follows that in the plastic regime that same property equals $x_{si}^v + x_{si}^p$. We adopt the foregoing.

In this work, we use the expression of Syamlal et al. (1993) for the solid viscous shear viscosity:

$$\mu_{si}^v = \frac{\phi_s s_i \rho_{si} \sqrt{\Theta_{si} \pi}}{6(3 - e_{ij})} \left[1 + \frac{2}{5} (1 + e_{ij}) (3e_{ij} - 1) \phi_s g_i \right] + \frac{4}{5} \phi_s^2 \rho_{si} s_i g_i (1 + e_{ij}) \sqrt{\frac{\Theta_{si}}{\pi}} \quad (4.17)$$

and the viscous dilatational (or granular bulk viscosity) viscosity is that of Lun et al. (1984):

$$\kappa_{si}^v = \frac{4}{3} \phi_s^2 s_i \rho_{si} g_i (1 + e_{ij}) \sqrt{\frac{\Theta_{si}}{\pi}} \quad (4.18)$$

where Θ_{si} is the solid granular temperature defined as one-third of the mean square velocity of the solid's random motion.

$$\Theta_{si} = \frac{1}{3} \langle v_{si}'^2 \rangle$$

where v_{si}' refers to the random fluctuating particle velocity. The granular temperature is governed by balance equations for the pseudointernal energies related to v_{si}' . This equation is given as (Ding and Gidaspow, 1990):

$$\frac{3}{2} [\partial_t (\rho_{si} \phi_s \Theta_{si}) + \nabla \cdot (\rho_{si} \phi_s \Theta_{si} \mathbf{u}_{si})] = \nabla \cdot (k_{\Theta_s} \nabla \Theta_{si}) + \mathbf{S}_{si} : \nabla \mathbf{u}_{si} - \gamma_{\Theta_i} + \phi_{ik} \quad (4.19)$$

where the left hand side of the equation represents the transport equation for the fluctuating energy with a source term right hand side. The first term on the right hand side is the diffusion of energy and k_{Θ_s} is the diffusion co-efficient, γ_{Θ_i} is the collisional dissipation energy and ϕ_{ik} is the energy exchange between i -th solid phase and a k -th solid phase or fluid phase. The diffusion co-efficient for granular energy k_{Θ_s} is given by Syamlal et al. (1993):

$$k_{\Theta_s} = \frac{15 \phi_s s_i \rho_{si} \sqrt{\Theta_{si} \pi}}{(41 - 33\eta)} \left[1 + \frac{12}{5} \eta^2 (4\eta - 3) \phi_s g_i + \frac{16}{15\pi} (41 - 33\eta) \eta \phi_s g_i \right] \quad (4.20)$$

where

$$\eta = 1/2(1 + e_{ij}) \quad (4.21)$$

the collisional dissipation energy and $\gamma_{\Theta i}$ which is the rate of energy dissipation within the i -th solid phase due to collisions between particles. This term is modelled using the expression of Lun et al. (1984):

$$\gamma_{\Theta i} = \frac{12(1 - e_{ij}^2)g_i}{s_i\pi^{1/2}}\rho_{si}\phi_s^2\Theta^{1.5} \quad (4.22)$$

ϕ_{ik} is given by Gidaspow et al. (1991):

$$\phi_{ik} = -3F_{ij}\Theta_{si} \quad (4.23)$$

where F_{ij} is the momentum exchange co-efficient between phase (solid or liquid) i and phase (solid) j . In this work, we adopt the algebraic expression for the granular temperature which neglects diffusion and convection in the transport equation in Eq. 4.19 (in other words, the production and dissipation of granular energy is in equilibrium), therefore we may write:

$$0 = S_{si} : \nabla \mathbf{u}_{si} - \gamma_{\Theta i} + \phi_{ik} \quad (4.24)$$

The normal viscous solid pressure P_{si}^v is based on the kinetic theory concepts and given by the expression of Syamlal et al. (1993) which is an extension of Lun et al. (1984):

$$P_{si}^v = 2\rho_{si}(1 + e_{ij})\phi_s^2 g_i \Theta_{si} \quad (4.25)$$

In the plastic regime, the solid stress is modelled using arbitrary functions which permits a certain level of compressibility in the solid phase. A closure for the plastic solid pressure which is frequently adopted by modellers is of the form:

$$P_{si}^p = \phi_{si} P^* \quad ; \quad P^* \equiv A(\phi_s - \phi_g)^B \quad (4.26)$$

However, we shall ignore this term as it leads to big pressure fluctuations and large numerical instabilities that could result in the simulation crashing. This is because A and B have typical high values ($A = 10^{25}$ and $B = 10$) causing huge deviations in ϕ_s and ϕ_g .

For the plastic shear viscosity μ_{si}^p we adopt the expression of Schaeffer (1987):

$$\mu_{si}^p = \frac{P_{si}^v \sin \zeta}{2 \sqrt{I_2(\mathbf{D}_i)}} \quad ; \quad I_2(\mathbf{D}_i) \equiv \frac{1}{2}[(Tr \mathbf{D}_i)^2 - Tr \mathbf{D}_i^2] \quad (4.27)$$

where ζ is the angle of internal friction of the i -th solid phase and $I_2(\mathbf{D}_i)$ refers to the second invariant of the strain tensor $\mathbf{D}_i$.

4.1.2 DQMOM Multi-Fluid Model

The polydisperse solid phase is modelled by a density function $f_n(\mathbf{x}, s, t)$ which defines the particle size distribution and whose transport equation is given by the PBE derived in chapter 2 and also reported here in equation 4.28.

$$\partial_t f_n(\mathbf{x}, s, t) + \nabla_{\mathbf{x}} \cdot [f_n(\mathbf{x}, s, t) \langle \mathbf{v} | s \rangle] = h_n(\mathbf{x}, s, t) \quad (4.28)$$

where s is the particle size, $\mathbf{x}$ is the position vector and t is time. In the expression above, h_n is the ‘generation’ term representing discontinuous jumps in size space (i.e. aggregation and breakage). Using DQMOM, the NDF, $f_n(\mathbf{x}, s, t)$, can be replaced by the quadrature approximation given by a summation of dirac delta functions:

$$f_n(\mathbf{x}, s, t) \approx \sum_{i=1}^v n_i(\mathbf{x}, t) \delta[s - s_i(\mathbf{x}, t)] \quad (4.29)$$

where n_i is the weight of the delta function corresponding to the characteristic particle size or node s_i . By substituting equation 4.29 into 4.28 one obtains the transport equations for the weights n_i and weighted nodes $n_i s_i$ written as:

$$c_i^n(\mathbf{x}, t) \equiv \partial_t n_i + \nabla_{\mathbf{x}} \cdot (n_i \mathbf{v}_i) \quad ; \quad c_i^s(\mathbf{x}, t) \equiv \partial_t (n_i s_i) + \nabla_{\mathbf{x}} \cdot (n_i s_i \mathbf{v}_i) \quad (4.30)$$

where $c_i^n(\mathbf{x}, t)$ and $c_i^s(\mathbf{x}, t)$ can be found from a linear system of the first $2v$ moments written in matrix form. The rigorous derivation of the transport equations for the weight and weighted nodes is given in § 2.5.1. The source terms $c_i^n(\mathbf{x}, t)$ and $c_i^s(\mathbf{x}, t)$ are defined via a linear system involving the first $2v$ moments (e.g. $k = 0, 1, 2, \dots, 2v - 1$) of the PSD. The linear system generated can be written in matrix form as:

$$\mathbf{X} \mathbf{c} = \mathbf{z}$$

where $\mathbf{X}$ is a $2v \times 2v$ matrix, $\mathbf{c}$ is a column matrix with its elements being the source terms $[c_1^n, c_2^n, \dots, c_v^n, c_1^s, c_2^s, \dots, c_v^s]^T$ and can be found by simply inverting $\mathbf{X}$ where $\mathbf{z}$ is also a column matrix with its elements being the moments of the generation term $[\mathcal{M}_0(h_n), \mathcal{M}_1(h_n) \dots \mathcal{M}_{2v-1}(h_n)]^T$.

For $v = 2$,

$$\mathbf{X} = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ -s_1^2 & -s_2^2 & 2s_1 & 2s_2 \\ -2s_1^3 & -2s_2^3 & 3s_1^2 & 3s_2^2 \end{bmatrix}$$

and the source terms are derived by inverting as shown:

$$\begin{bmatrix} c_1^n \\ c_2^n \\ c_1^s \\ c_2^s \end{bmatrix} = \frac{1}{(s_1 - s_2)^3} \begin{bmatrix} (3s_1 - s_2)s_2^2 & -6s_1s_2 & 3(s_1 + s_2) & -2 \\ (s_1 - 3s_2)s_1^2 & 6s_1s_2 & -3(s_1 + s_2) & 2 \\ 2s_2^2s_1^2 & -(4s_1^2 + s_1s_2 + s_2^2)s_2 & 2(s_1^2 + s_1s_2 + s_2^2) & -s_1 - s_2 \\ -2s_2^2s_1^2 & (s_1^2 + s_1s_2 + 4s_2^2)s_2 & -2(s_1^2 + s_1s_2 + s_2^2) & s_1 + s_2 \end{bmatrix} \begin{bmatrix} \mathcal{M}_0(h_n) \\ \mathcal{M}_1(h_n) \\ \mathcal{M}_2(h_n) \\ \mathcal{M}_3(h_n) \end{bmatrix}$$

If the matrix is singular, then it becomes impossible to invert. To overcome this, a small pertubation can be added to the nodes to make $\mathbf{X}$ full rank. This method however, tends to fail as the number of nodes increases. Therefore, alternative approaches to solving the problem are proffered.

In this work we shall set the source vector $\mathbf{c}$ to be zero in the computational cells where singularity takes place.

For consistency with variables used in the gas-solid multi-fluid model, we relate the weights and nodes to the solid fraction ϕ_{si} and the weighted node $\phi_{si}s_i$ for each solid phase. The volume fraction of each solid phase is related to the node and weight thus:

$$\phi_{si} = k_v s_i^3 n_i \quad (4.31)$$

and the weighted node is:

$$\phi_{si}s_i = k_v s_i^4 n_i \quad (4.32)$$

where k_v is the volumetric shape factor (for spherical particles $k_v = \pi/6$). Using the relations in equations 4.31 and 4.32, we can write the transport equations as:

$$\partial_t(\phi_{si}\rho_{si}) + \nabla \cdot (\phi_{si}\rho_{si}\mathbf{u}_{si}) = 3k_v\rho_{si}s_i^2c_i^s - 2k_v\rho_{si}s_i^3c_i^n \quad (4.33)$$

$$\partial_t(\phi_{si}s_i\rho_{si}) + \nabla \cdot (\phi_{si}s_i\rho_{si}\mathbf{u}_{si}) = 4k_v\rho_{si}s_i^3c_i^s - 3k_v\rho_{si}s_i^4c_i^n \quad (4.34)$$

Reader is referred to Appendix B.1.1 and B.1.2 for the derivation of equations 4.33 and 4.34. Equation 4.33 represents the continuity equation for the i th solid phase in the presence of aggregation and breakage (compare with equation 4.3). The volume fraction of each phase will change according to the characteristic length s_i as a result of aggregation and breakage in order to imitate the evolution of the PSD. Equation 4.34 is solved in the multi-fluid model as a user-defined scalar. Therefore the DQMOM Multi-Fluid model for the polydisperse solid phase consists of the equations 4.5, 4.33 and 4.34. Also, we need to account for the change in the PSD due to aggregation and breakage. Therefore, h_n in Eq. 4.28 needs to be related to known expressions of aggregation and breakage from the theory of population balances.

It is also important to note that previous validation studies of DQMOM carried out by Marchisio et al. (2003b), Marchisio et al. (2003c) and Marchisio and Fox (2003) have demonstrated that by using two or three nodes, the lower-order moments of the PSD are tracked with surprisingly small errors and that the quadrature approximation actually mimics the the evolution and shape of the investigated PSD (McGraw, 1997). The DQMOM model describes the model used by Fan et al. (2004) in their work. For QMOM model implemented in Fluent for our work, please refer to section 2.5.2.

4.1.3 Aggregation and Breakage Models

Our investigation considers changes in PSD due only to aggregation and breakage for which we can express h_n as:

$$h_n(\mathbf{x}, s, t) = h_n^b(\mathbf{x}, s, t) + h_n^a(\mathbf{x}, s, t) \quad (4.35)$$

where $h_n^b(\mathbf{x}, s, t)$ and $h_n^a(\mathbf{x}, s, t)$ are the breakage and aggregation source terms respectively and are defined as follows:

$$h_n^b(\mathbf{x}, s, t) = \int_s^{+\infty} b(\mathbf{x}, r, t) f_n(\mathbf{x}, r, t) P(s, |\mathbf{x}, r, t) dr - b(\mathbf{x}, s, t) f_n(\mathbf{x}, s, t) \quad (4.36)$$

$$h_n^a(\mathbf{x}, s, t) = \frac{1}{2} \int_0^s a[\mathbf{x}, u(s, r), r, t] f_n[\mathbf{x}, u(s, r), t] f_n(\mathbf{x}, r, t) \frac{\partial u}{\partial s} dr - \\ f_n(\mathbf{x}, s, t) \int_0^{+\infty} a(\mathbf{x}, s, r, t) f_n(\mathbf{x}, r, t) dr \quad (4.37)$$

where $a(\mathbf{x}, s, r, t)$, $b(\mathbf{x}, r, t)$ and $P(s, |\mathbf{x}, r, t)$ are the aggregation kernel, breakage kernel and daughter distribution function respectively. The moment transform of the source term h_n is given as (details in § 3.2.4):

$$\mathcal{M}_k(h_n) = (1/2) \sum_{i=1}^v \sum_{j=1}^v n_i n_j s^k(s_i, s_j) a(s_i, s_j) - \sum_{i=1}^v \sum_{j=1}^v n_i n_j s^k(s_i) a(s_i, s_j) + \sum_{i=1}^v n_i b(s_i) P_i^k - \sum_{i=1}^v n_i s^k(s_i) b(s_i) \quad (4.38)$$

We shall use the aggregation and breakage kernels derived from the kinetic theory of granular flow and the binary breakage daughter distribution functions already introduced in chapter three.

$$a(s_i, s_j) = \Psi_a g_{ij} \left(\frac{3\theta_s}{\rho_s} \right)^{\frac{1}{2}} (s_i + s_j)^2 \left(\frac{1}{s_i^3} + \frac{1}{s_j^3} \right)^{\frac{1}{2}} \quad (4.39)$$

$$b(s_i) = \Psi_b \sum_{j=1}^v n_j g_{ij} \left(\frac{3\theta_s}{\rho_s} \right)^{\frac{1}{2}} (s_i + s_j)^2 \left(\frac{1}{s_i^3} + \frac{1}{s_j^3} \right)^{\frac{1}{2}} \quad (4.40)$$

$$P_i^{(k)} = r^k \frac{m^{k/3} + n^{k/3}}{(m+n)^{k/3}} \quad (4.41)$$

With these approximations, the source term h_n is closed.

Chapter 5

Results and Discussions

In this chapter, we shall proceed to replicate the work of Fan et al. (2004) using the QMOM approach. It is worth mentioning that the work of Fan et al. (2004) was based on DQMOM. In theory, QMOM and DQMOM are equivalent. Based on this equivalence, we shall carry out a preliminary validation of our model by comparing our results with the results of Fan et al. (2004).

5.1 Model Implementation

We employed the commercial CFD code Fluent to run the simulations. With the use of user-defined functions (UDFs) and sub-routines we were able to implement the QMOM, aggregation and breakage source terms as well as post-processing within Fluent. We tracked the evolution of the first four, six and eight integer moments of the NDF, thus defining three, four and five phases respectively in the multi-fluid model. In each of these cases, one of the phases represents the fluid phase. The quadrature weights and nodes (referring to the number densities and particle sizes respectively) were derived from the moments via the PD algorithm. The transport equations for the NDF moments were added to the code as user-defined scalars but not without some simple manipulations of the default user-defined scalar (UDS) equation in Fluent.

In Fluent, the user defined scalar is associated with either the disperse phase or the mixture phase. In the first case, Fluent solves the following equation:

$$\partial_t(\phi_k \mathcal{M}_k) + \nabla_{\mathbf{x}} \cdot (\phi_k \mathcal{M}_k \mathbf{v}_k) = \mathcal{S}_k(\mathbf{x}, t) \quad (5.1)$$

whereas in the second case Fluent solves:

$$\partial_t(\rho_m \mathcal{M}_k) + \nabla_{\mathbf{x}} \cdot (\rho_m \mathcal{M}_k \mathbf{v}_m) = \mathcal{S}_k(\mathbf{x}, t) \quad (5.2)$$

where

$$\rho_m \equiv \phi_g \rho_g + \rho_s \sum_{s=1}^v \phi_s \quad ; \quad \rho_m \mathbf{v}_m \equiv \phi_g \rho_g \mathbf{u}_g + \rho_s \sum_{s=1}^v \phi_s \mathbf{v}_s \quad (5.3)$$

The above equations are different from the equation we propose to solve i.e.

$$\partial_t \mathcal{M}_k + \nabla_{\mathbf{x}} \cdot (\mathcal{M}_k \mathbf{v}_k) = \mathcal{S}_k(\mathbf{x}, t)$$

where

$$\mathbf{v}_k(\mathbf{x}, t) = \frac{1}{\mathcal{M}_k(\mathbf{x}, t)} \sum_{i=1}^v n_i(\mathbf{x}, t) s_i^k(\mathbf{x}, t) \mathbf{v}_i(\mathbf{x}, t)$$

We simply replaced $\mathbf{v}_k$ in (5.1) with the expression for the velocity of the k -th moment of the distribution as defined above using the `DEFINE_UDS_FLUX` macro (Fluent, 2009) available in Fluent (the reader is referred to the Fluent UDF manual). In addition, the user defined scalar was defined as $\mathcal{M}_k / \phi_k$, so that this ratio is multiplied by the volume fraction thus reducing the equation to the correct one which we intend to solve. The PD algorithm was implemented using the `DEFINE_ADJUST` macro. Other multi-fluid models described in chapter 4 are available in Fluent as default and as such did not require implementation via UDFs. The breakage and aggregation source terms were implemented using the `DEFINE_SOURCE` macro in Fluent. For each of the user defined equations we included two source terms: one for aggregation and another for breakage. Fluent adopts the finite-volume discretization scheme and we employed the pressure-based solver recommended for low-speed incompressible flows. At every time step we used a maximum of 150 iterations and an average time step to 1×10^{-4} s for most of the simulations.

5.2 Implementation Check

In order to check that the model has been implemented properly within Fluent, we try duplicating the results of Fan et al. (2004). In their work, they employed both constant and kinetic kernels. Five cases were investigated, three with the constant kernels and two using the kinetic kernels. They investigated no-aggregation-no-breakage, aggregation-dominant and breakage-dominant cases for the constant kernel. For the kinetic kernels, they only tested breakage-dominant and aggregation dominant cases. We shall present the results starting from the constant kernel to the kinetic kernel.

5.2.1 Constant Kernel

In the first set of simulations, we employed the constant aggregation and breakage kernels. Thus the aggregation kernel, $a(s_i, s_j)$ and the breakage kernel $b(s_i)$ were assumed to be independent of particle diameter, velocity and other properties. Values of the model parameters used are listed in Table 5.1.

Three cases were investigated. In *Case 1*, the aggregation and breakage kernels were both set to zero. *Case 2* is aggregation dominant while *Case 3* is breakage dominant.

In *Case 1*, setting both the aggregation and breakage kernels to zero indicates that no aggregation and breakage are present. We expect that the volume-average mean particle size d_{32} (refer to Appendix C.1.1 for the passages leading to the mathematical definition of d_{32}), namely the ratio between the third moment m_3 and the second moment m_2 of the PSD, to remain constant. This is because the third order moment and the second order

Table 5.1: Initial values of particle diameter(s_i) and solid-phase volume fractions (ϕ_{si})

	Nodes	$i = 1$	$i = 2$	$i = 3$	$i = 4$
particle diameter s_i (μm)	2	183	356		
	3	174	263	409	
	4	171	225	316	420
phase volume fraction, ϕ_{si}	2	0.274	0.356		
	3	0.196	0.229	0.205	
	4	0.157	0.157	0.157	0.157

$$^* m_0 = 100873.042\text{cm}^{-3} \quad m_1 = 2103.820\text{cm}^{-2} \quad m_2 = 47.740\text{cm}^{-1} \quad m_3 = 1.204$$

$$m_4 = 3.4 \times 10^{-2}\text{cm} \quad m_5 = 1.07 \times 10^{-3}\text{cm}^2$$

moment corresponding to the volume and surface area respectively remain unchanged. Let us first consider the results obtained with the two-node and the three-node quadrature approximations by using the first order upwind discretization scheme. Figures 5.1 and 5.2 report the profiles of the quadrature nodes and volume fractions for two-node quadrature approximation at the start of the simulation and at pseudostationary conditions. The simulation based on the two-node and three-node quadrature formula ran smoothly. Figure 5.3 shows the graph of d_{32} against time for 2-node and 3-node quadrature approximation for first order upwind discretization scheme. Our results show a good agreement with the work of Fan et al. (2004). Similar results using the second order upwind discretization scheme were obtained for the two-node quadrature approximation but for the three-node quadrature approximation we observed a decrease in d_{32} with time. In the case of the latter, we suspected that the wrong prediction may have been due to corruption of the independently transported moments of the PSD. We do not expect that numerical diffusion should be a problem in this case. However, to rule out all possible doubts, we have gone ahead to check whether numerical diffusion plays a role or not in the wrong prediction.

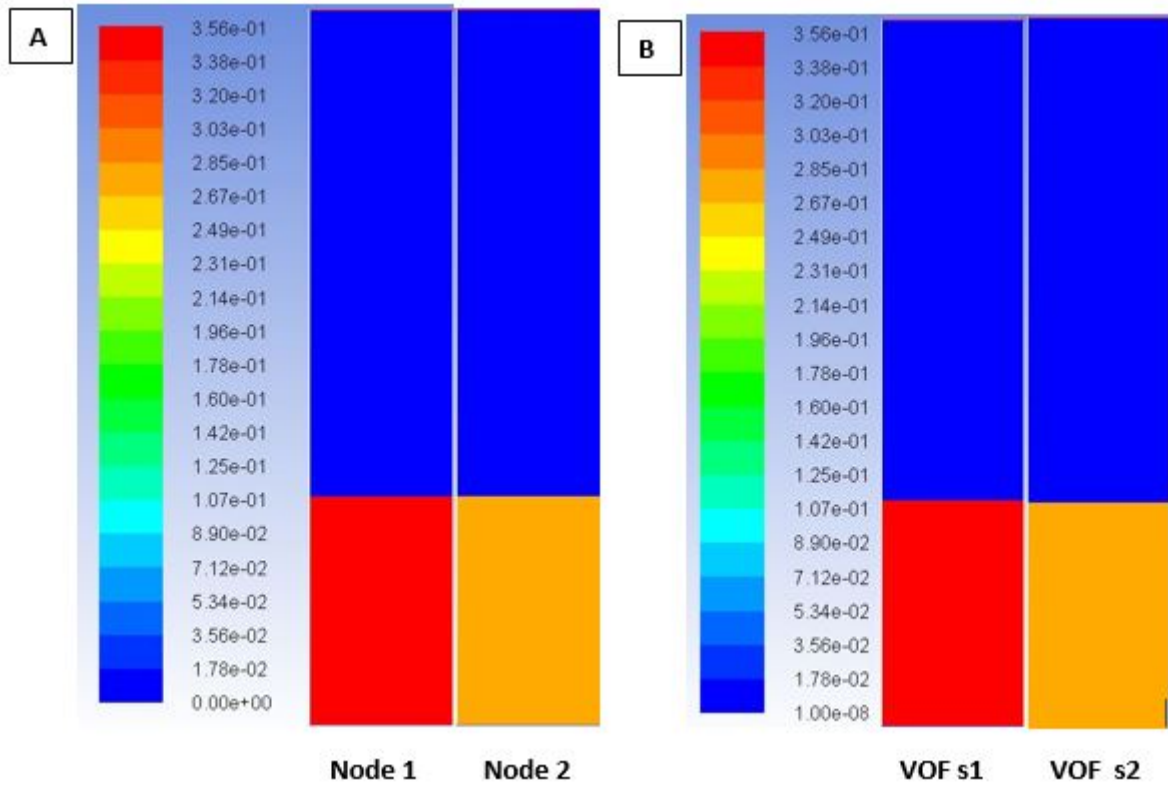


Figure 5.1: [A] Nodes and [B] Volume fractions for Case 1 at time 0s for $N = 2$

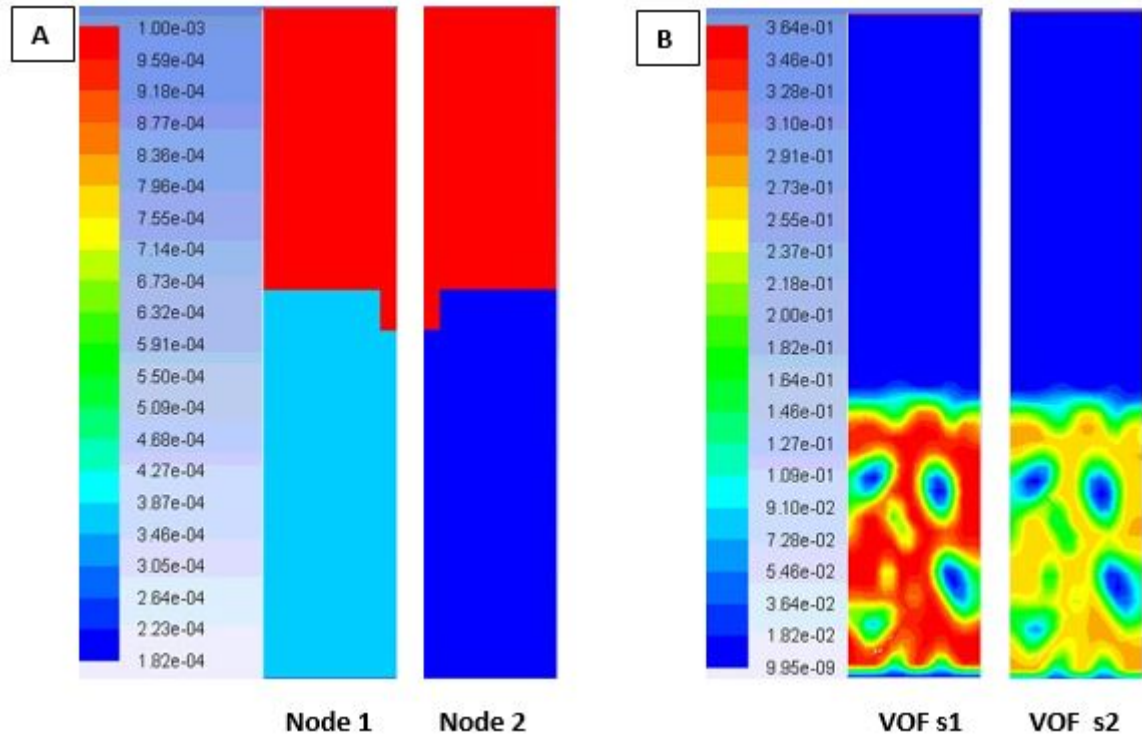


Figure 5.2: [A] Nodes and [B] Volume fractions for Case 1 at time 10s for $N = 2$

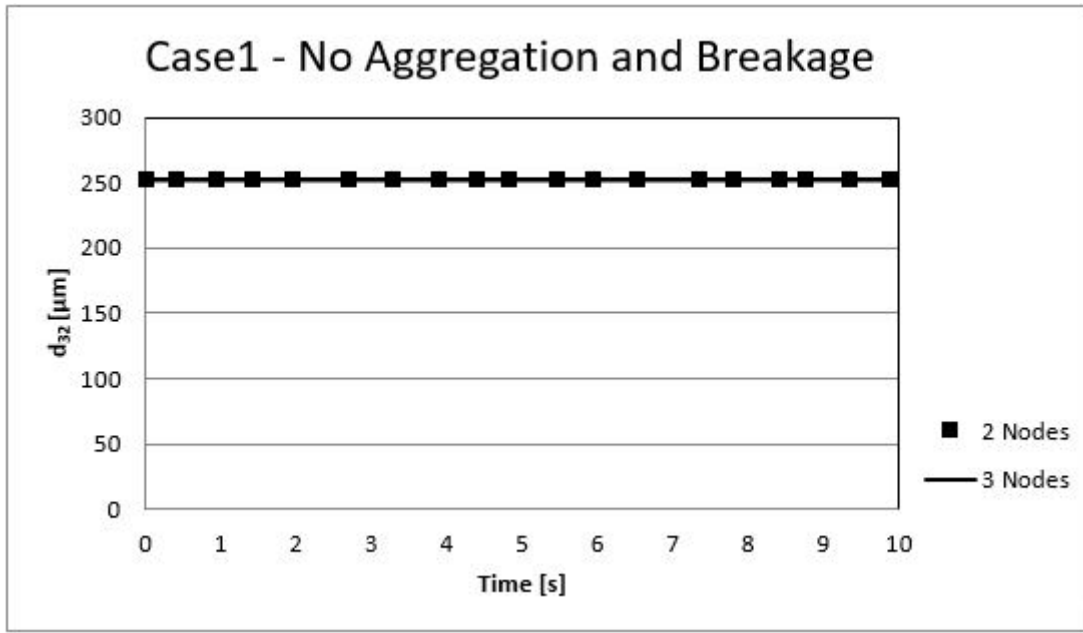


Figure 5.3: volume-average mean particle size vs. time for Case 1 and $N = 2$ and 3

Numerical diffusivity depends on the discretization scheme and the computational grid employed. We cannot eliminate numerical diffusion as it is always present when one integrates purely convective equations with CFD codes, however we can minimize numerical diffusivity by reducing the computational grid and employing higher order discretization schemes. We therefore employed a 5mm x 5mm computational grid to re-run the three-node quadrature approximation using second-order discretization. This did not change the results significantly. Hence, the implementation of a code to check for presence of moment corruption.

For the four node quadrature formula, we observed moment corruption while using both first- and second- order upwind schemes before the simulation eventually crashed at about 0.4s. The corrupted moments caused the values of d_{32} to decrease over time instead of staying constant. Also we observed that as the number of nodes increased, so did the complexity of computation within Fluent, run time and the simulation sensitivity to the choice of time step size.

When we couple the CFD solution with the PBE through the solution of moment transport equations, the calculated moments have a tendency to form invalid sequences as a result of numerical simplifications made to solve the moment transport equations. The conventional transport algorithms treat each moment independently from one another and cannot guarantee to preserve inter-relations within members of a moment set, especially when using high order discretization schemes (Acher et al., 2013; Mazzei et al., 2012; McGraw, 2012; Petitti et al., 2010; Wright, 2007). When the moment transport equations are solved independently, the resulting moments obtained through the inversion procedure might not preserve the relationships, leading to invalid moment sets (McGraw, 2006; Wright, 2007). A valid moment set contains crucial information about a physical distribution, whereas, an invalid moment sequence does not represent any physical distribution.

We therefore employed two methods to check the validity of the moment sequence. The first method involved constructing a difference table. By taking the natural logarithm of the raw moments obtained from the simulation, an i -th order difference column can be constructed. The convexity requirement, which is only a necessary condition for valid moment sequence, is satisfied if and only if the second-order differences are non-negative (the reader is referred to McGraw (1997, 2012) and Appendix D1.1 for more details of the method). Once the moment satisfies the convexity condition, which is a necessary condition for the moment sequence validity, the moment sequence is then subjected to the positive alpha sequence enforcement (PASE) test, which forms the sufficient condition. The PASE convexity check consists of “alpha sequences” which are mathematical quantities introduced by Gordon (1968) that are related to the Hankel-Hadamard determinant sequence (refer to McGraw (2006) and Appendix D1.2 for detail of the PASE check). Figure 5.4 shows the results of the “difference-check” and the “PASE-check” for the three-node quadrature simulation at pseudostationary condition using the second-order discretization scheme. As we can see, the variable of the “difference-check” is equal to one every where in the computational domain meaning that the moment sequence generated from the simulation at 0.05s pass the necessary convexity check. However when subjected to a more rigorous PASE check, we observe that the computational domain have values equal to one and zero. The value of zero within the bed shows that the convexity requirement was not satisfied. Hence, not satisfying the sufficient condition for valid moment set.

We intend to implement correction algorithms within Fluent to try overcome the problem of moment corruption.

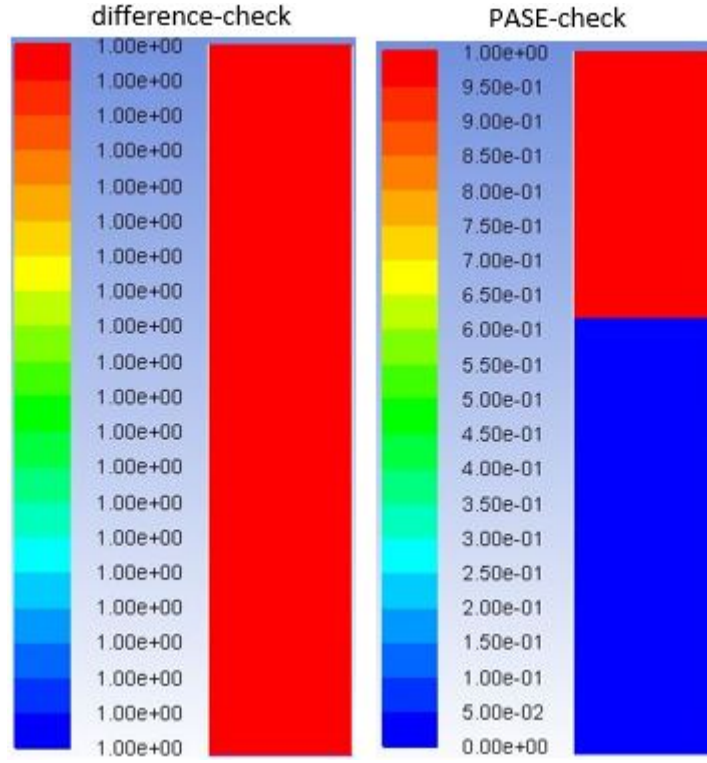


Figure 5.4: difference- and PASE- checks for three-node quadrature approximation using second-order upwind discretizaion scheme at 0.05s - Case 1

For Case 2, we set the aggregation kernel to $1 \times 10^{-5} m^3/s$ and the breakage kernel to $0.1 s^{-1}$. Here, aggregation is dominant and the particles become larger, so the volume-average mean particle size increases with time. For the two-quadrature node simulation using both first- and second- order upwind schemes, we observed d_{32} increased with time (see Figure 5.5) while the value of the NDF moments decreased except for the third order moment which stayed constant for the duration of the simulation. Figure 5.7 shows the volume-average normalized moments for Case 2 using two-node quadrature formula. The normalized moments are calculated by dividing the volume-average moments by their values at time $t = 0s$ (refer to appendix C for passages leading to the mathematical definition of $\bar{m}_k$).

$$M_k(t) = \frac{\bar{m}_k(t)}{\bar{m}_k(0)}$$

Some moments have particular meaning. For example, m_0 represents the total particle number density, whereas m_2 relates to the total particle area, and m_3 is related to the total particle volume. In Figure 5.7, the expected effects of aggregation are observed: the total number of particles indicated by m_0 decreases, as do m_1 and m_2 , whereas the total particle volume m_3 remains constant.

For the three-quadrature node simulations, we obtained similar results as in the case for two-node quadrature formula (see Figure 5.6). However, we observed the values of d_{32} decreased with time as a result of moment corruption when we ran the simulation using the second-order upwind scheme. Likewise, we observed similar moment corruption when we ran the simulation for the four-node quadrature formula.

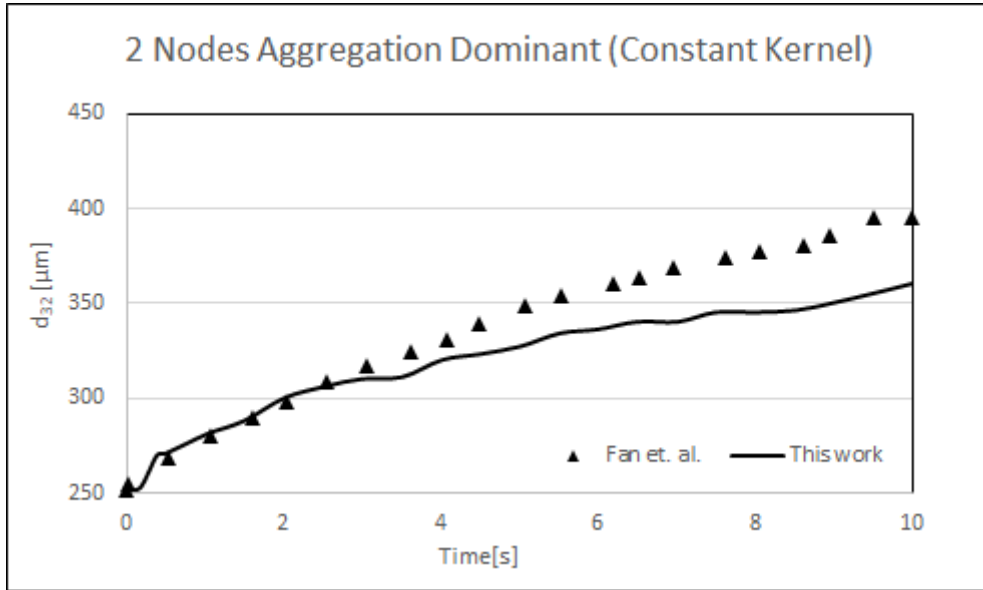


Figure 5.5: volume-average mean particle size vs time for Case 2, second order upwind scheme (2 Nodes)

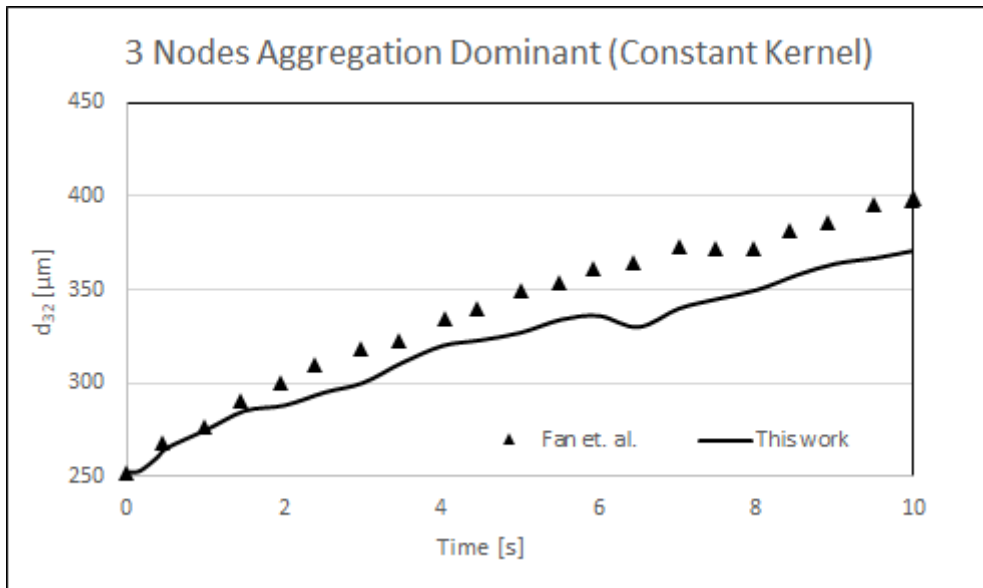


Figure 5.6: volume-average mean particle size vs time for Case 2, first order upwind scheme (3 Nodes)

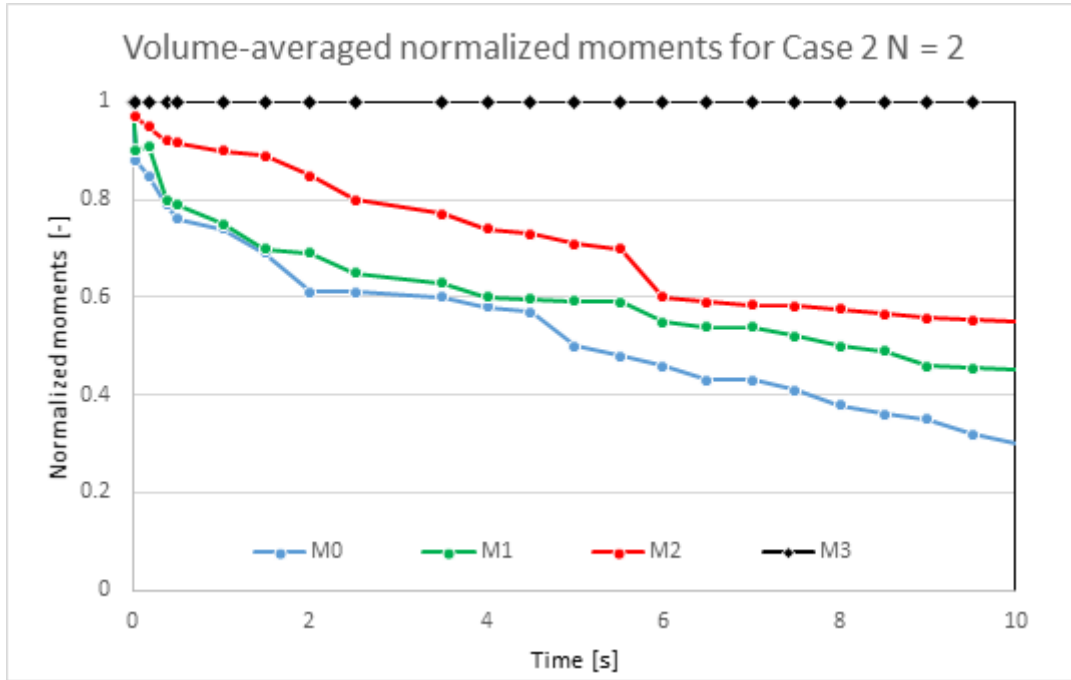


Figure 5.7: volume-average normalized moments for case 2 for two-node quadrature formula

We compare the particle size and volume fractions at the middle of the bed based on the first four and six NDF moments. In Tables 5.2 and 5.3, we report the particle size and the volume fractions at pseudostationary condition at 10s. In order to compare with the values from Fan et al. (2004), we calculated the first eight moments from the results they presented and using the PD algorithm implemented in MATLAB, we back-calculated the nodes and weights for three- and two- quadrature formula using the first six and four moments respectively. From the weights of the quadrature formula we were able to determine the volume fractions.

Table 5.2: PSD at the middle of the bed at time 10s for Case 2 using N=3

<i>Fan et. al. (2004)</i>					
$s_1(\mu m)$	$s_2(\mu m)$	$s_3(\mu m)$	$\phi_1(-)$	$\phi_2(-)$	$\phi_3(-)$
52	284	643	0.0150	0.2432	0.3797
<i>This work</i>					
$s_1(\mu m)$	$s_2(\mu m)$	$s_3(\mu m)$	$\phi_1(-)$	$\phi_2(-)$	$\phi_3(-)$
94	294	631	0.0208	0.2510	0.3600

Table 5.3: PSD at the middle of the bed at time 10s for Case 2 using N=2

<i>Fan et. al. (2004)</i>			
$s_1(\mu m)$	$s_2(\mu m)$	$\phi_1(-)$	$\phi_2(-)$
225	539	0.1137	0.5097
<i>This work</i>			
$s_1(\mu m)$	$s_2(\mu m)$	$\phi_1(-)$	$\phi_2(-)$
215	567	0.1112	0.5310

We observe that the predictions of the particle sizes and volume fractions obtained from both works differ quantitatively. However, we observe that in terms of trends, the results are similar. As we can see in Figures 5.5 and 5.6, the results mimic those of Fan et al. (2004) up till about 2s and then begin to deviate. Fan et al. (2004) stated in their work that “the simulation results found using other polydisperse models may differ quantitatively from those reported here”. In other words, modifications of models, for example the drag formulation or the stress tensors can alter the quantitative replication of the results. In this work we have used the commercial CFD code Fluent with constitutive equations differing from those of MFIx used in the work of Fan et al. (2004). For example, the granular temperature and the radial distribution formulation in MFIx completely differ from those in Fluent. Therefore transitioning from MFIx to Fluent may have caused the disparities observed in the quantitative values for the particle sizes and volume fractions.

For Case 3, we set the aggregation and breakage kernels to $1 \times 10^{-5} m^3/s$ and $1 s^{-1}$. Breakage is dominant and the particles reduce in size, so the volume-average mean particle size decreases with time. For the two-quadrature node simulation using both first- and second-order upwind schemes, we observed d_{32} decreased with time (see Figure 5.8) while the value of the NDF moments increased except for the third order moment which stayed constant for the duration of the simulation. For the three-quadrature node simulations, we obtained similar results as in the case for two-node quadrature formula (see Figure 5.9). Tables 5.4 and 5.5 show the particle sizes and volume fractions at the middle of the fluidized bed for three- and two- quadrature formula respectively.

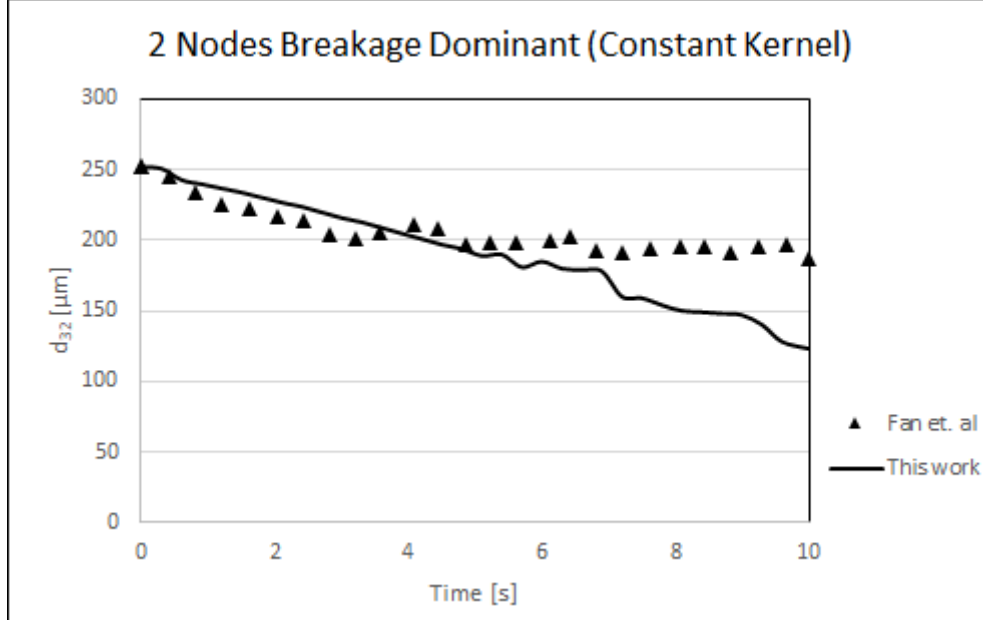


Figure 5.8: volume-average mean particle size vs time for Case 3, second order upwind scheme (2 Nodes)

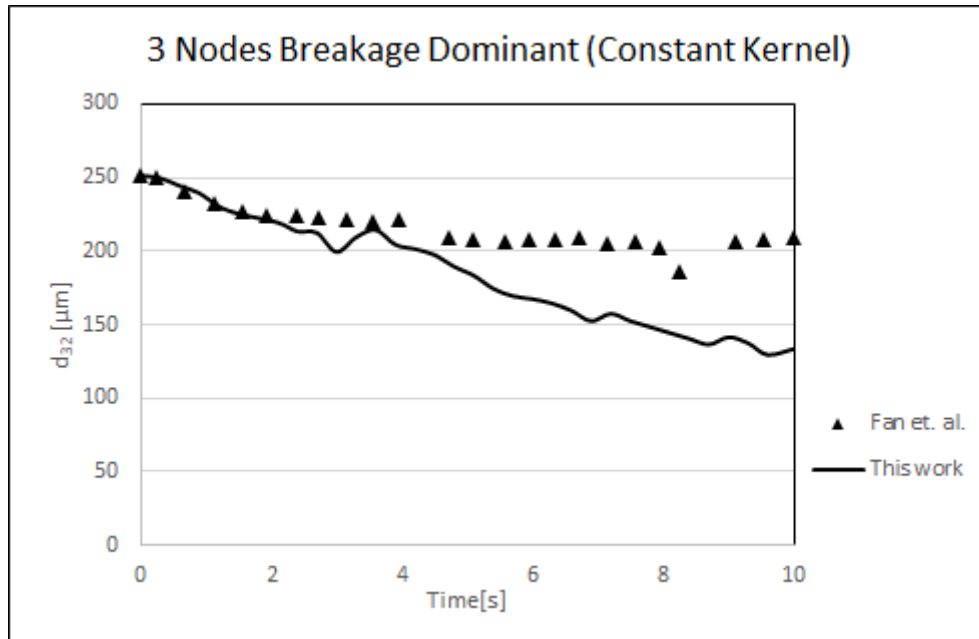


Figure 5.9: volume-average mean particle size vs time for Case 3, first order upwind scheme (3 Nodes)

However, we observed the values of d_{32} decreased rapidly to about $94\mu\text{m}$ after 0.5s when we ran the three-node quadrature simulation using second order upwind scheme. We observed that the large deviation in the result obtained was due to the corruption of the moments which we will try to correct using correction algorithms implemented in Fluent.

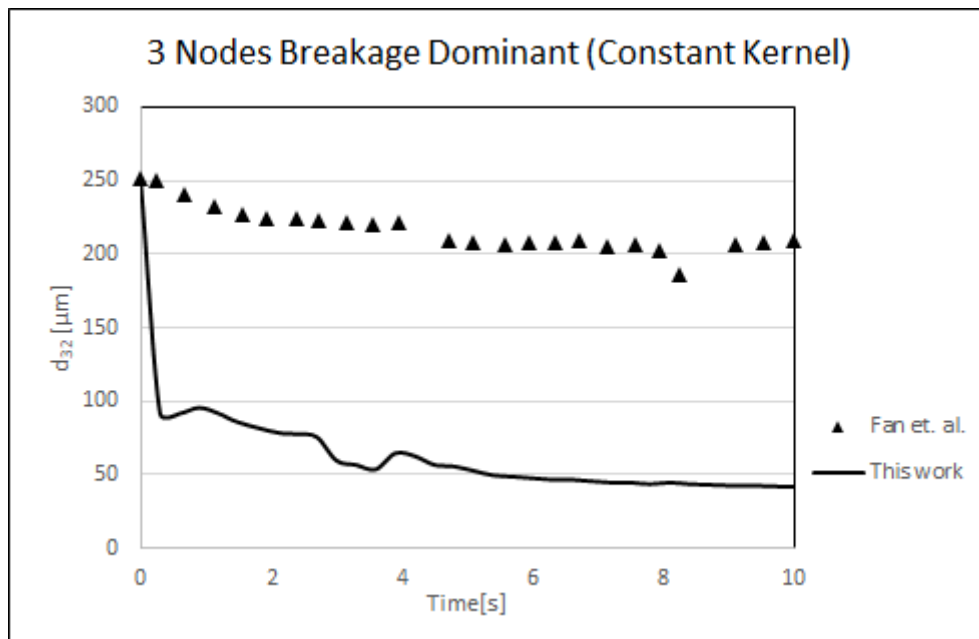


Figure 5.10: volume-average mean particle size vs time for Case 3, second order upwind scheme (3 Nodes)

Table 5.4: PSD at the middle of the bed at time 10s for Case 3 using N=3

<i>Fan et. al. (2004)</i>					
$s_1(\mu m)$	$s_2(\mu m)$	$s_3(\mu m)$	$\phi_1(-)$	$\phi_2(-)$	$\phi_3(-)$
27	62	110	0.0128	0.1912	0.1449
<i>This work</i>					
$s_1(\mu m)$	$s_2(\mu m)$	$s_3(\mu m)$	$\phi_1(-)$	$\phi_2(-)$	$\phi_3(-)$
38	84	124	0.0300	0.1510	0.1721

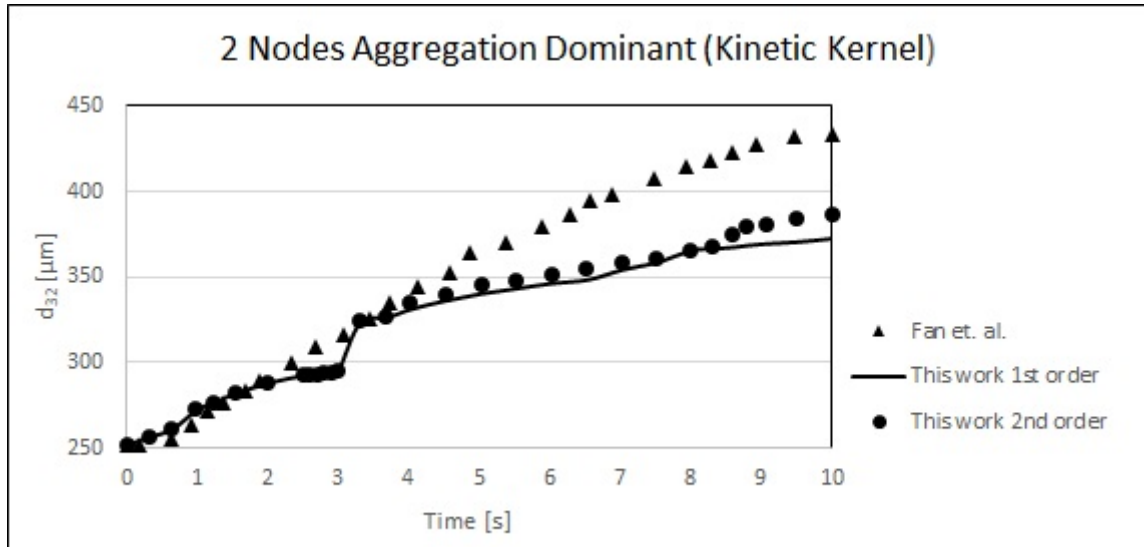
Table 5.5: PSD at the middle of the bed at time 10s for Case 3 using N=2

<i>Fan et. al. (2004)</i>			
$s_1(\mu m)$	$s_2(\mu m)$	$\phi_1(-)$	$\phi_2(-)$
34	85	0.0424	0.3068
<i>This work</i>			
$s_1(\mu m)$	$s_2(\mu m)$	$\phi_1(-)$	$\phi_2(-)$
29	72	0.0321	0.4910

5.2.2 Kinetic Kernel

Two cases were investigated for the kinetic kernel model. *Case 4* is aggregation dominant and *Case 5* is breakage dominant.

Figures 5.11 and 5.12 show how the volume-average mean particle size d_{32} for Case 4 varies with time for two- and three- node quadrature approximations respectively. It is possible to observe, that our results show a similar trend with that of Fan et al. (2004) but differ quantitatively as already observed in the cases involving constant kernels. The latter is expected for reasons already explained such as the different constitutive equations employed in MFIX and ANSYS Fluent.

**Figure 5.11:** volume-average mean particle size vs time for Case 4, 1st and 2nd order upwind scheme (2 Nodes)

For example the radial distribution by Iddir and Arastoopour (2005) which we employed in Fluent represents a realistic densely packed bed when the total volume fraction of the solids approach the maximum packing limit. First, the particles aggregate, particles increase in size and the rate of aggregation reduces as the particles increase in size which is expected. The aggregates tend to move to the bottom of the bed causing possible defluidization as aggregation continues. As defluidization sets in, the granular temperature θ_s approaches zero thereby stopping further aggregation from occurring. Figure 5.13 shows the contour of the granular temperature at 10s for the two dispersed phases. We observe from figure 5.13 that the granular temperature at the bottom of the bed approaches zero while the value is significant at the top of the bed. In this case, whereas aggregation stops at the bottom of the bed due to defluidization, it continues at the top thereby reducing the rate of aggregation. We also observe that the graph in Figure 5.11 shows a further decrease in the slope from 9s to show the decrease in aggregation as the process proceeds. In contrast, Case 2 which is aggregation dominant using the constant kernel continues aggregating irrespective of defluidization because there is no variable in the kernel to trigger the on-set of defluidization at the bottom of the vessel.

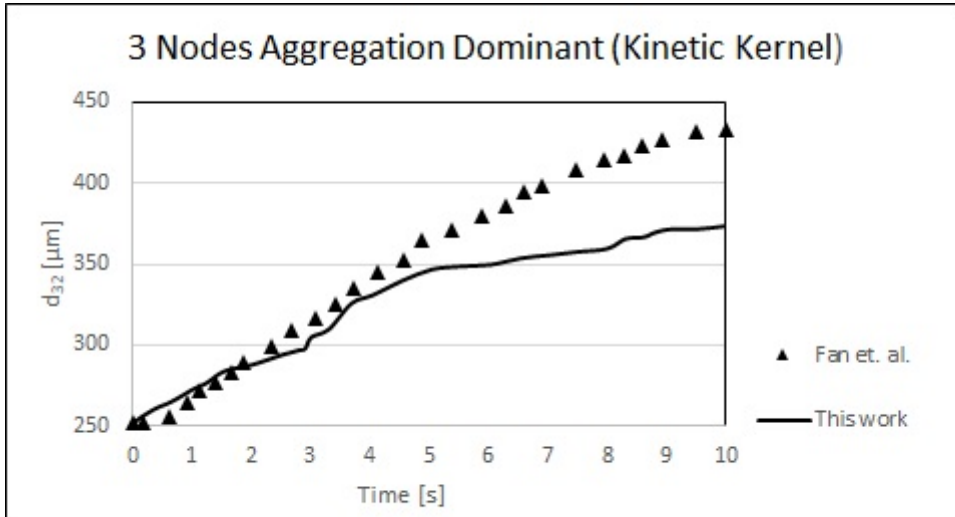


Figure 5.12: volume-average mean particle size vs time for Case 4, first order upwind scheme (3 Nodes)

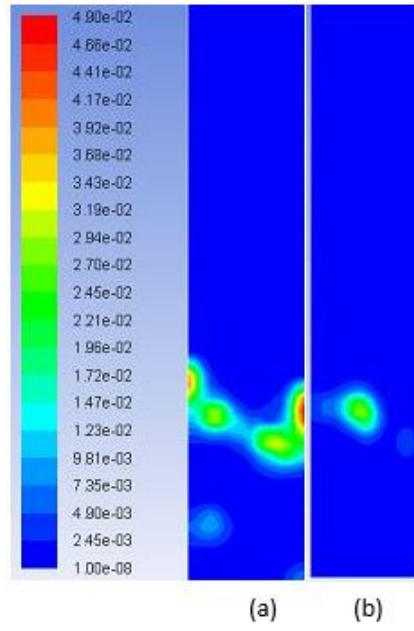


Figure 5.13: contour of granular temperature for (a)solid 1 (b) solid 2 at 10s for Case 4 (2 Nodes)

Results for Case 5 are reported in Figure 5.14. We observe that generally the the volume-average mean particle diameter decreases with time. In this case we notice that the results obtained using both two- and three- nodes are very similar to those obtained by Fan et al. (2004) with differences in the quantitative values of d_{32} especially after 4s. We observe in Figure 5.15 that the value of d_{32} decreases rapidly within 1s when case simulation was run using a four-node quadrature formula. This is as a result of corruption of the transported moments which was observed in previous simulations using the four-node quadrature formula.

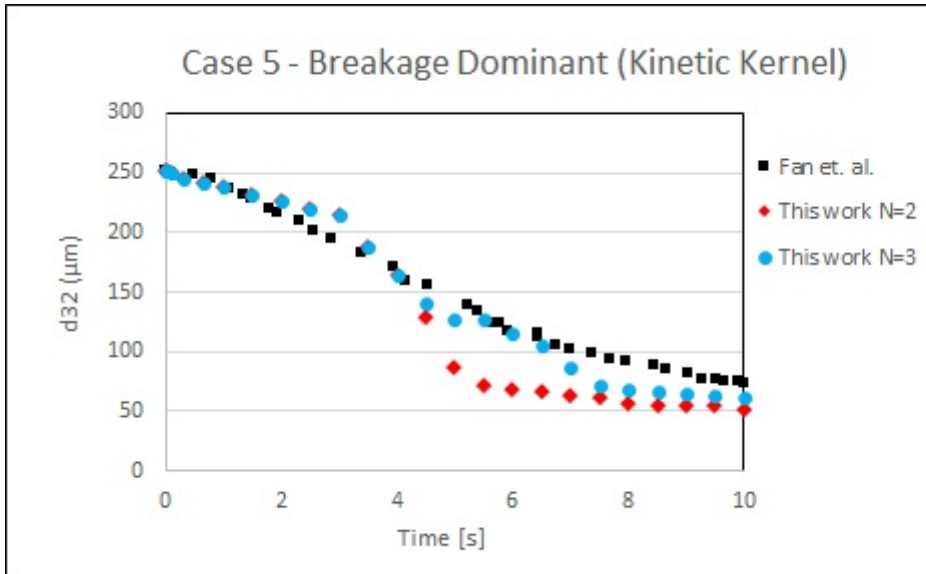


Figure 5.14: volume-average mean particle size vs time for Case 5

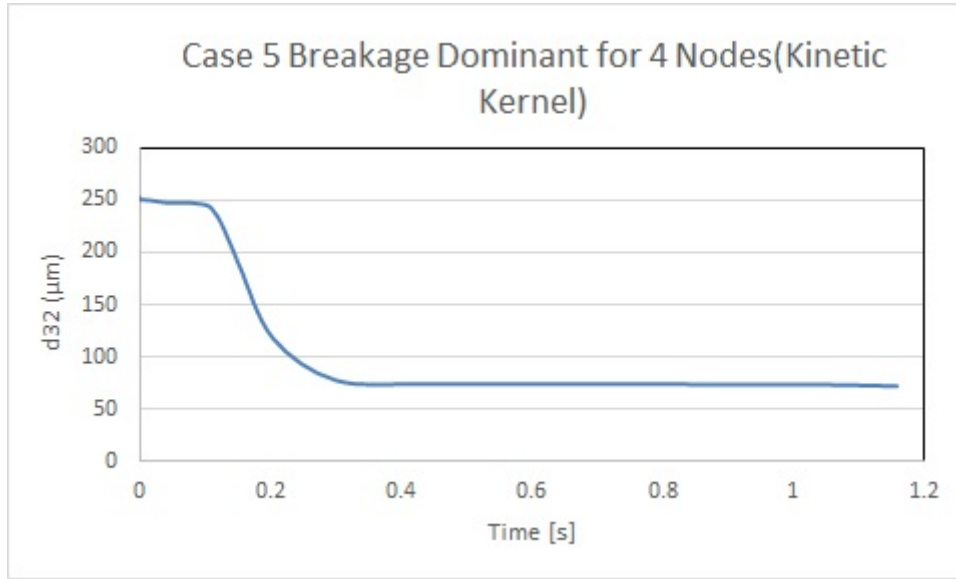


Figure 5.15: volume-average mean particle size vs time for Case 5 using $N=4$

5.3 Correcting Corrupted Moments

We reported the problem of moment corruption especially for higher order upwind scheme simulations and for simulations tracking more than six NDF moments. When the CFD is coupled with the PBE through the solution of moment transport equations, the calculated moments have a tendency to form invalid sets because in QMOM, the moment transport equations are solved independently. Hence, the resulting moments obtained via the inversion procedure might not preserve the relationships between the moments which make them part of a valid set. A valid moment set provides crucial information about a physical distribution, whereas an invalid sequence does not represent any valid physical distribution. Vikas et al. (2010) specifically proposed the so-called “quasi-high-order” advection scheme for the QMOM to preserve realizability of moment sets. However, owing to the fact the QMOM model used in this work is built on an already existing numerical solver, modifying the available advection scheme is either impossible or not desired (Afzalifar et al., 2017). Therefore, we shall consider alternative solutions presented in the literature. We already introduced two ways of checking the validity of a moment sequence. Going forward, we intend to implement an algorithm to solve the problem of nonrealizable moment sequence while tracking more than six moments or using higher order discretization schemes with the quadrature method of moments.

One of the solutions reported in the literature is the McGraw (2006) correction procedure also referred to as the *filter method* (McGraw, 2012). Because moment corruption is likely as a result of improper assignment of one, or at most a few of the moments in the sequence, we would like to adjust only those moments. For a sufficiently long sequence the filter method is the way to achieve this goal. In this method a difference table (see Appendix D.1.1) is constructed and used to identify the order of the moment that is corrupted and then provide an optimal correction by minimizing the sum of the squared differences of the second order differences, which through correction ensures the smooth-

ness of $\ln(m_k)$ sequence. This correction method by McGraw (2006) only takes account of the convexity condition, which is only necessary but not a sufficient condition. The convexity condition is satisfied if and only if all the elements of the second-order difference vector are non-negative. In some cases, the convexity test is passed but the moment set is still unphysical. In this case we will need to subject the moments to a test that identifies both necessary and sufficient conditions such as the positivity of the Hankel-Hadamard determinants constructed from the moments.

In place of the Hankel-Hadamard determinants, we will use another set of non-negative quantities, the alpha sequence (see Appendix D.1.2), investigated by Gordon and generated by him using the P-D algorithm (Gordon, 1968), to investigate the validity of the moment set.

Wright (2007) replaced nonrealizable moment set using a log-normal distribution or the arithmetic average of two log-normal distributions by retaining the zeroth and third moments from the original moment sequence. For a log-normal distribution, a generic moment can be written as:

$$m_k = N_T \exp \left(k\mu + \frac{k^2\sigma^2}{2} \right) \quad (5.4)$$

where N_T is the number density, μ and σ correspond to the mean and variance of the PSD.

We intend to use a combination of these methods to achieve correction of corrupted moments as shown in Figure 5.16 :

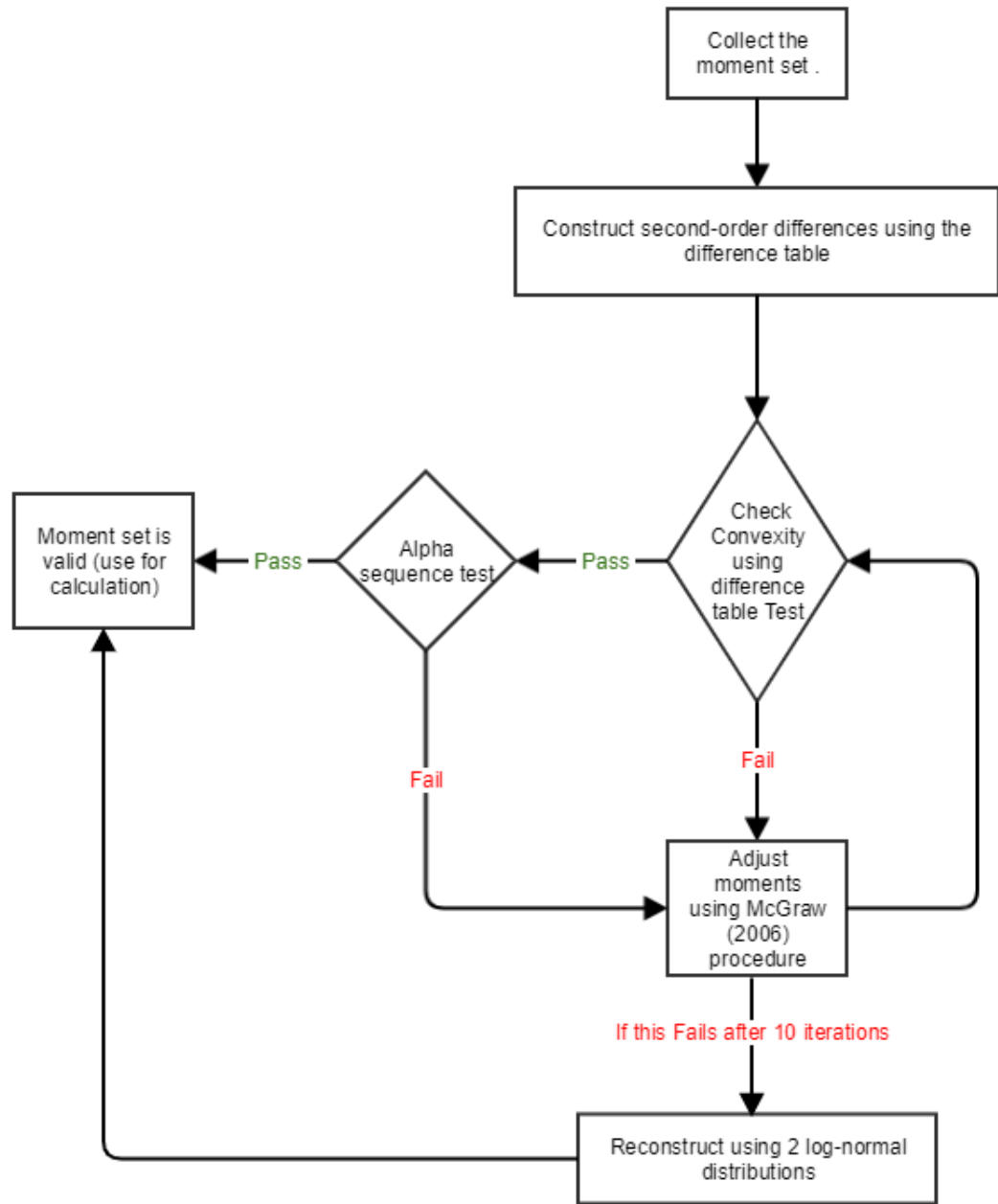


Figure 5.16: Flowchart to show how the invalid moments are identified and corrected

5.3.1 Implementation of Moment Correction Algorithm

The correction algorithm was implemented within Fluent using the DEFINE_ADJUST macro. The moments calculated in Fluent were retrieved and a difference table was constructed in $\ln_k$. We defined the response vectors $\mathbf{b}_k$ according to McGraw (2006) moment correction procedure (reader is referred to text for full reference and description of procedure). The moment sets are then tested for convexity of the function $\ln_k$ using both the difference table and the positive alpha sequence enforcements. If the moment set fails both or any of the validity tests then the McGraw moment correction is applied. In cases

where the moment set cannot be corrected in a reasonable amount of iterations, we have limited the number of iterations to 10 after which the recommendation of Wright (2007) is applied.

5.3.2 Results from Moment Correction

So we ran simulations which had previously crashed or produced wrong results with the moment correction algorithm implemented. We present the results here.

We ran simulations with the implemented correction algorithm for Case 1 using the four-quadrature node which had initially shown a decrease in d_{32} . The results showed a constant value of d_{32} for about 0.6s before gradually decreasing in value. The simulation eventually crashed at about 1.7s compared to 0.4s without the moment correction algorithm (See Figure 5.17).

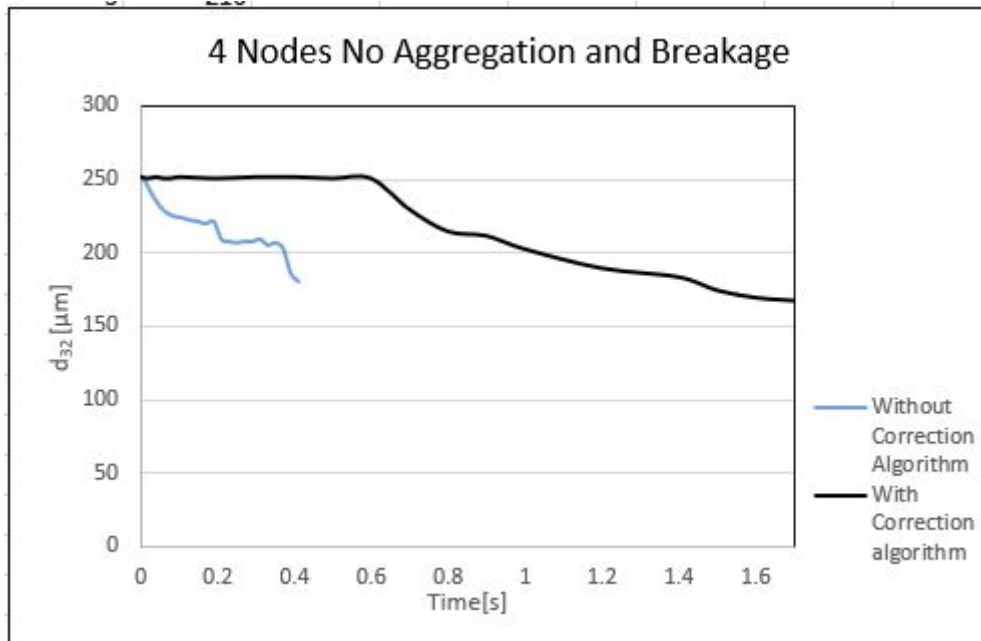


Figure 5.17: Graph to show how d_{32} varies with time for Case 1 with moment correction algorithm

Similar to results observed in Case 1, we noticed a delay in the time it takes the simulations to crash after the correction algorithm was implemented for Case 2 to Case 5 for four quadrature nodes. The algorithm also failed to curtail divergence while using the second order upwind for three quadrature nodes for Cases 2 to 5.

In order to be sure the correction algorithm was correctly implemented, we implemented it in the default ANSYS Fluent QMOM model. However, we were only able to run Case 1, Case 2 and Case 3 because of the restrictions in modifying the aggregation and breakage kernels in the default QMOM fluent model to represent the kinetic aggregation and breakage models we have chosen for this work. The default QMOM model differs from the model we had initially implemented in that it does not afford the phases to be transported with different velocities. The implication of this is that the particles move with the

same velocity and hence cannot freely segregate. We only present second order upwind results because from the literature, second order upwind discretization schemes for simulating QMOM-PBE models are more prone to moment corruption.

First, we will run a test case without the correction algorithm after which we will run the same simulation with the correction algorithm implemented in every control volume of the computational domain at every solution iteration. We see from Figure 5.18 that without the correction algorithm, the two-node and three-quadrature gave us the required trend we expected. However, with the four-node quadrature, we experienced a similar trend as in the case with different velocities: decreasing values of d_{32} with time. In this case, we did not experience any crash in simulation for the four-node quadrature simulation even with moment corruption leading to decrease in the value of d_{32} . When we ran the same simulation with the correction algorithm implemented, we noticed that we were able to capture the expected trend where d_{32} remains constant throughout the simulation for two-, three-, and four-quadrature nodes.

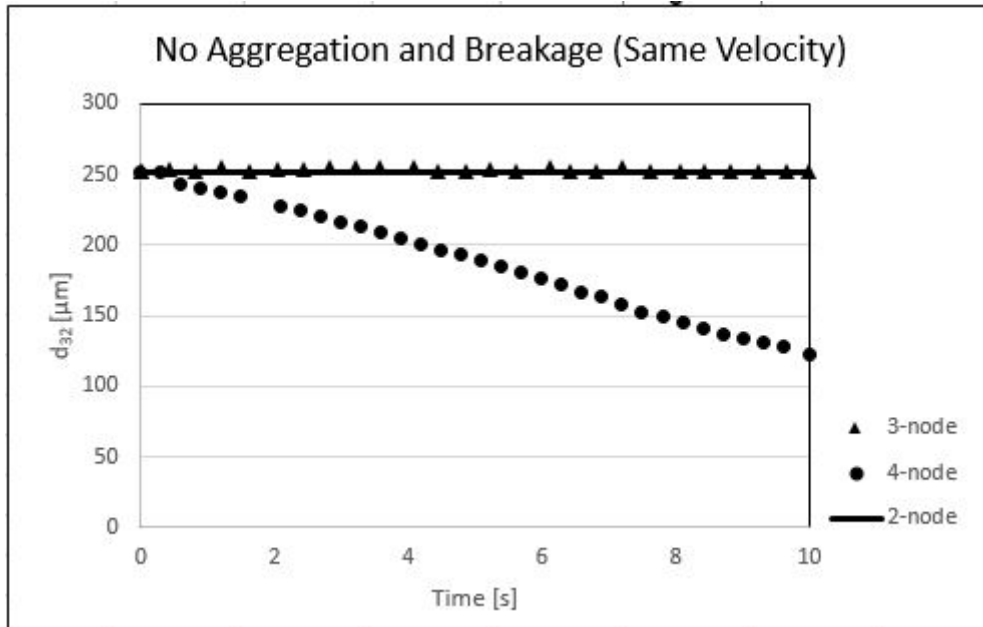


Figure 5.18: Graph to show how d_{32} varies with time for Case 1 without moment correction algorithm

For Case 2 and Case 3, we observed that simulations involving two-quadrature nodes were void of moment corruption while simulations with three- and four- quadrature nodes showed moment corruption which affected the results and eventually led to the simulations crashing. We can see in Figure 5.19 negative results for the values of the volume fraction for Node 1 for the four-node quadrature simulation at 0.91s for Case 2. This indicates the presence of moment corruption leading to inaccurate results. Figures 5.20 and 5.22 show the results for the three- and four- node quadrature results for Case 2 and Case 3 respectively with the moment correction implemented.

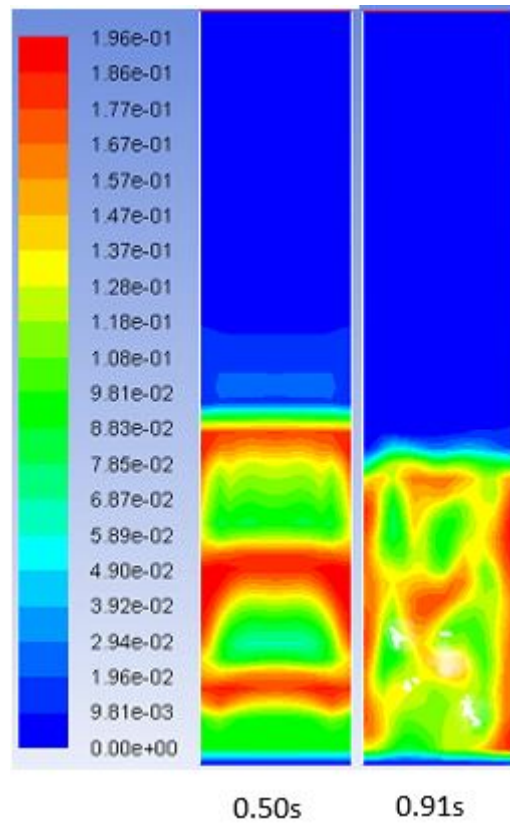


Figure 5.19: Contour of volume fraction of Node 1 for four-node quadrature Case 2 at 0.50s and 0.91 s. (White contours show the presence of moment corruption in the affected cells)

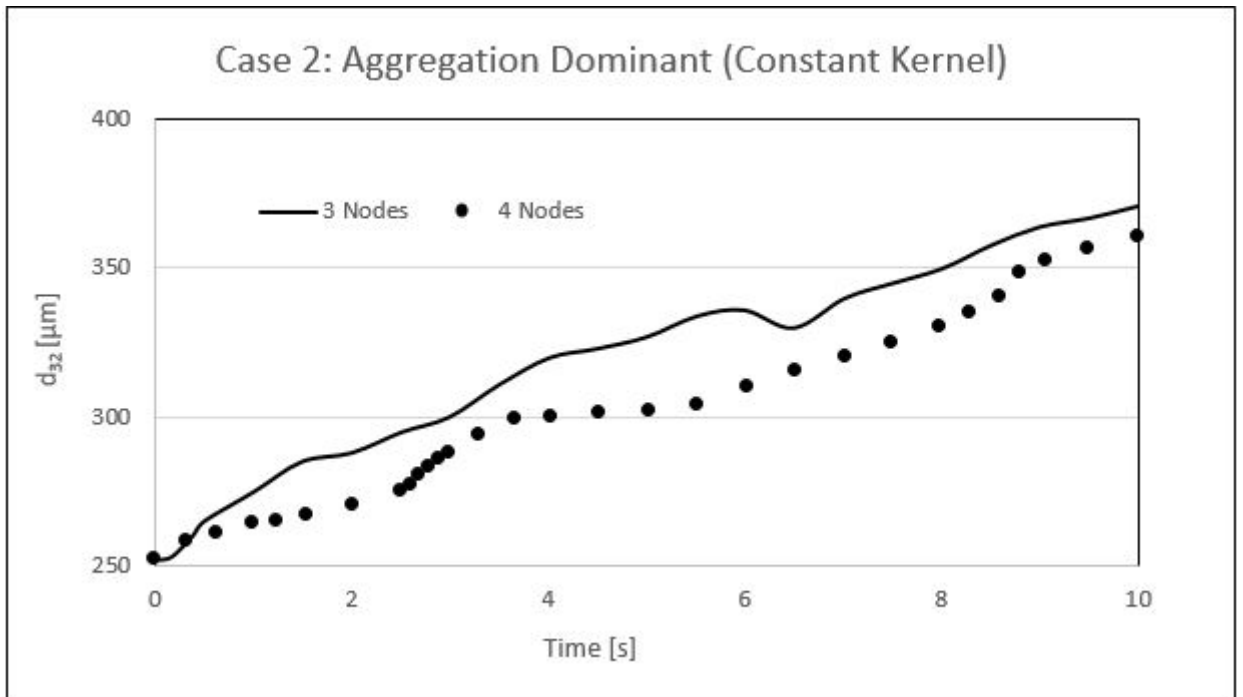


Figure 5.20: Graph to show how d_{32} varies with time after implementing moment correction for Case 2 simulations (particles transported at same velocity)

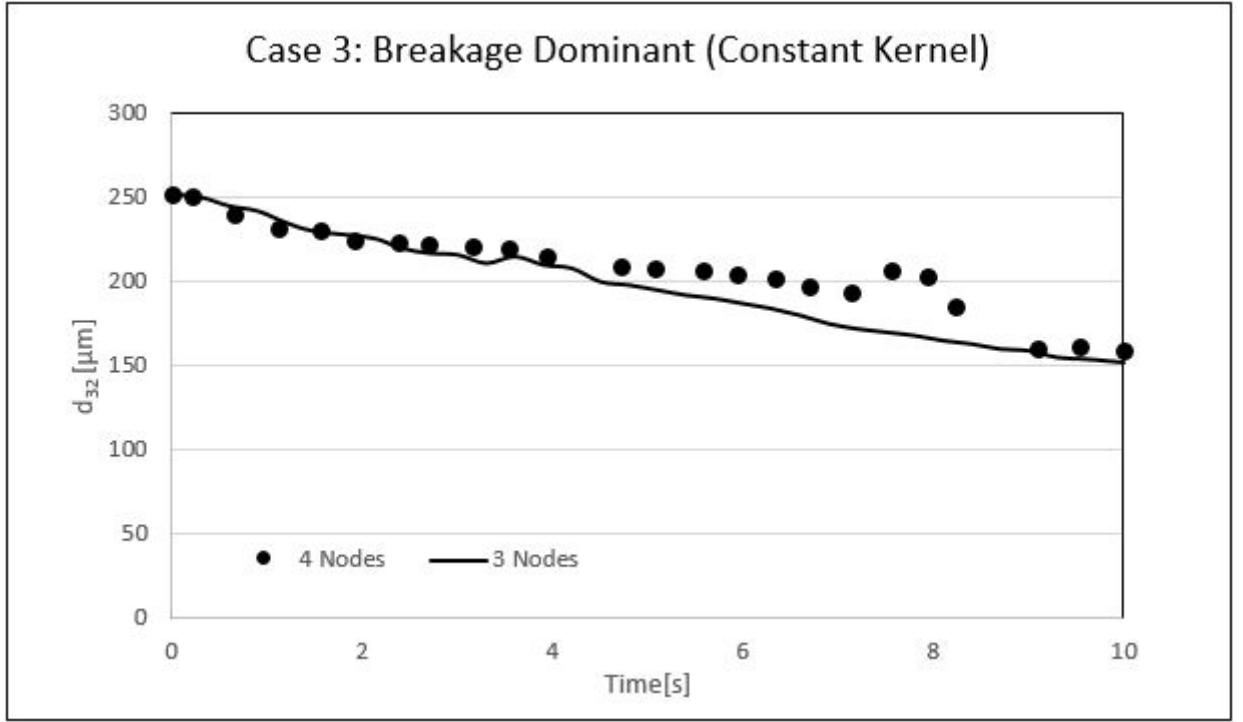


Figure 5.21: Graph to show how d_{32} varies with time after implementing moment correction for Case 3 simulations (particles transported at same velocity)

We have been able to establish that the correction algorithm was implemented correctly and that QMOM models with the quadrature nodes transported with same velocity are easily corrected compared to similar simulations involving quadrature nodes transported with different velocities.

From the literature, we mostly see simulations involving quadrature nodes with same velocities. There is no available literature, to the best of our knowledge, with a successfully functional correction algorithm for QMOM models with different velocities for the quadrature nodes. We suspect that this feature in the model allowing the quadrature nodes to be transported at different velocities may have contributed to the complication of correcting the moments. We take a closer look at the two models: the QMOM model with the quadrature nodes transported with same velocity and that with different velocities for the quadrature nodes.

$$\partial_t \mathcal{M}_k + \nabla_{\mathbf{x}} \cdot (\mathcal{M}_k \mathbf{v}_m) = \mathcal{S}_k(\mathbf{x}, t) \quad (5.5)$$

$$\partial_t \mathcal{M}_k + \nabla_{\mathbf{x}} \cdot (\mathcal{M}_k \mathbf{v}_k) = \mathcal{S}_k(\mathbf{x}, t) \quad (5.6)$$

where

$$\mathbf{v}_k(\mathbf{x}, t) = \frac{1}{\mathcal{M}_k(\mathbf{x}, t)} \sum_{i=1}^v n_i(\mathbf{x}, t) s_i^k(\mathbf{x}, t) \mathbf{v}_i(\mathbf{x}, t) \quad (5.7)$$

We see from equation 5.7 that the velocity at which the individual moments are transported is indirectly related to the value of the moments, where n_i and s_i^k are the weight

and nodes respectively derived from the moment $\mathcal{M}_k$ via the product-difference algorithm. This implies that a corrupt moment will replicate ‘corrupt’ velocity value which in turn complicates the solution. In contrast, the mixture model which Fluent uses in solving equation 5.5 does not relate to the moment. Fluent uses the relative (slip) velocity, $v_i - v_g$ (v_i being the velocity of the secondary phase and v_g the velocity of the primary phase) to prescribe an algebraic relation for the relative velocity, with the basic assumption that a local equilibrium between the phases is reached over a short spatial length scales. The consequence of this is that the resulting solution is simplified in terms of the velocity without amplifying the moment corruption problem.

5.4 Segregation of Fluidized Powders

One important aspect of modelling is that of checking for robustness of the model. In this case, we ask if the model is able to capture the segregation pattern exhibited by fluidized powders at low fluidization velocities. Mixing and segregation play a very important role in fluidization technology. The difference in particle sizes makes segregation possible. The literature is rich with articles on segregation behaviours of fluidized powders differing in size and density. However, the segregation patterns of these particles were mostly reported from the perspective of binary mixtures, that is, bidisperse powders differing in size and/or density. In these studies (Marzocchella et al., 2000; Nienow et al., 1987; Formisani et al., 2001; Van Wachem et al., 2001; Cooper and Coronella, 2005), the smaller and lighter particles referred to as the *flotsam* accumulates at the top of the bed while the relatively bigger and heavier particles referred to as the *jetsam* tend to migrate to the bottom part of the bed. Here, we want to approach segregation studies from the perspective of continuous PSD.

To investigate segregation using the QMOM model, we will compare our results with the experimental results of Mazzei (2008). The system comprise two superposed powders with different PSDs with the lower mean size particles placed beneath the particles having a higher mean size. This would mean that small and large particles were free to move to the top and bottom of the bed respectively. Reader is referred to Mazzei (2008) for details of the experiment apparatus and procedure. Figure 5.22 shows a diagrammatic representation of the experiment and Table 5.6 shows the PSD of the two powders A and B at the bottom and top of the bed respectively at $t = 0s$.

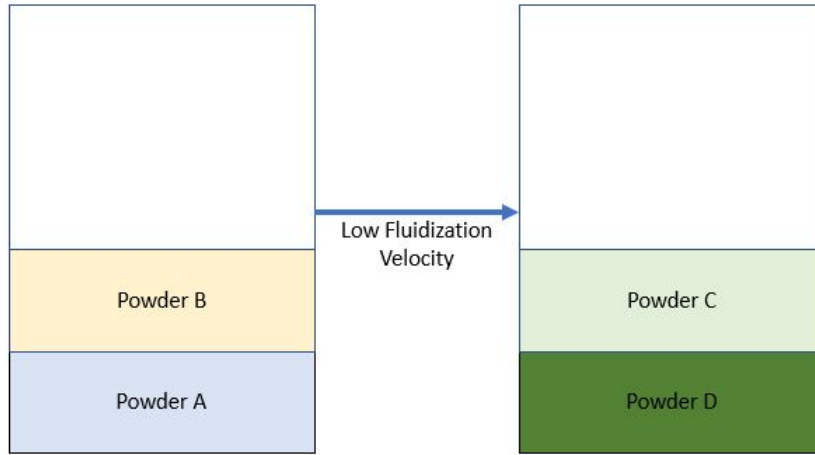


Figure 5.22: Diagrammatic representation of the segregation experiment

Table 5.6: Initial values of the quadrature nodes and weights obtained experimentally at $t = 0s$ with voidage of 0.4 (Mazzei et al., 2009)

<i>Powder A</i>					
$s_1(\mu m)$	$s_2(\mu m)$	$s_3(\mu m)$	$\phi_1(-)$	$\phi_2(\mu m)$	$\phi_3(-)$
61	89	112	0.069	0.400	0.131
<i>Powder B</i>					
$s_1(\mu m)$	$s_2(\mu m)$	$s_3(\mu m)$	$\phi_1(-)$	$\phi_2(\mu m)$	$\phi_3(-)$
208	287	395	0.143	0.376	0.081

5.4.1 Boundary and Initial Conditions

Two-dimensional bed with initial height of 300mm in a 600mm x 350mm vessel; Initial bed height for each of the powders were 150mm each with powder A beneath powder B as shown in Figure 5.22. The initial quadrature nodes and weights are summarised in Table 5.6. Void fraction of 0.4 was applied everywhere in the bed. A uniform inlet gas of 0.10 m/s was applied from the bottom of the fluidized bed, low enough to fluidize the powders but not high enough to cause mixing. A 5mm square cell computational grid was employed in the simulation. To run the simulations, we used Fluent 17.2. We ran simulations using the second-order upwind scheme. At each time step, we used a maximum of 100 iterations to compute the flow variables and tolerance was set to 10^{-5} . The time step was set to $10^{-4}s$ and under-relaxation factors of 0.20 were adopted for all the variables. The multifluid modelling approach is employed with closure relationships defined for the averaged equations of motion as reported in chapter 4. We divided the bed into five horizontal layers of equal heights in order to derive the axial segregation profiles.

5.4.2 Fluidized Bed Segregation Results

We simulated the fluidized beds under conditions that should promote segregation. As mentioned earlier, the system modelled is the same as that presented by Mazzei (2008) but we only considered a low fluidization velocity of 0.10 m/s; low enough to fluidize

the bed without causing mixing. In the experiment, the bed was divided into five layers of equal height and the PSD of each layer was analysed via sieving. From the PSD, the volume density function integer moments were calculated and the quadrature nodes and weights were obtained. The reader is again referred to Mazzei (2008) for details of the experiment.

We present the results of our simulations at pseudo-stationary conditions at 10s of running the simulation using the three-node quadrature method of moments. The bed was divided into five layers and the average NDF moments, weights and nodes were computed for each layer. The quadrature weights however were converted from number of particles to volume fraction using the relationship

$$\langle \phi_{sk} \rangle = k_v \langle s_k \rangle^3 \langle n_{sk} \rangle$$

We can see from Figure 5.23 that the computational values of the quadrature nodes are uniform everywhere within the bed. This differs a little bit from the experimental values which are also spatially uniform except for the bottom of the bed (layer 0 corresponds to the distributor plate while layer 5 represents the top layer of the bed) where the particles are slightly larger. We observe no more than 13% variation in the numerical values of the quadrature nodes from those obtained via experiment.

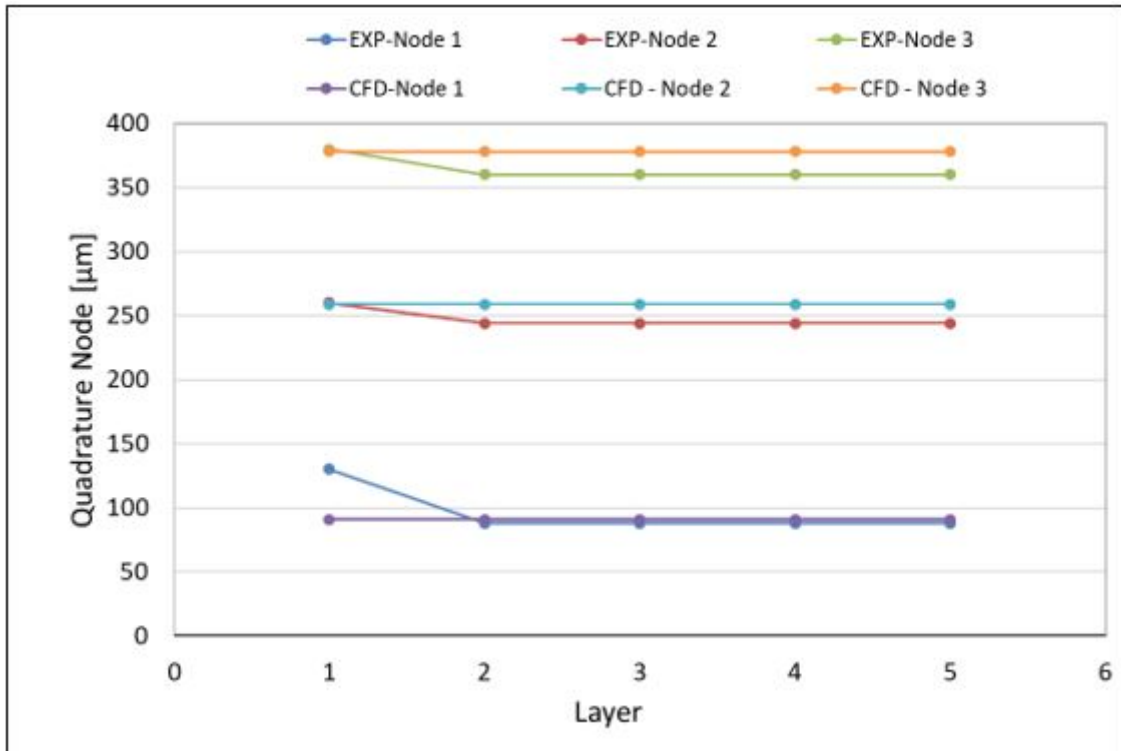


Figure 5.23: Quadrature Nodes for experimental and simulation results at pseudo-stationary conditions ($t = 0s$) for fluidization velocity = 0.10m/s

The experiment was a simple case of segregation not influenced by size-changing phenomena such as growth, aggregation and breakage. Segregation was purely as a result of

the particle size distribution and the fluidizing velocity. To this end, our expectation is that an axial non-uniform spatial profile of the volume fraction (derived from the quadrature weights) will exist in the bed if segregation takes place. This is exactly what we observe from Figure 5.24. We expect that quadrature node 3 being the largest in size and heaviest will decrease in number as we move from bottom to top of the bed. This is exactly what we observe from Figure 5.25 which shows the contour of volume fraction for quadrature node 3 at pseudo-stationary conditions. Figure 5.24 shows the axial segregation profile for quadrature node 3 at pseudo-stationary conditions. This shows a good agreement with experiment except for the top of the bed where quadrature node 3 is over-predicted by approximately about 25%.

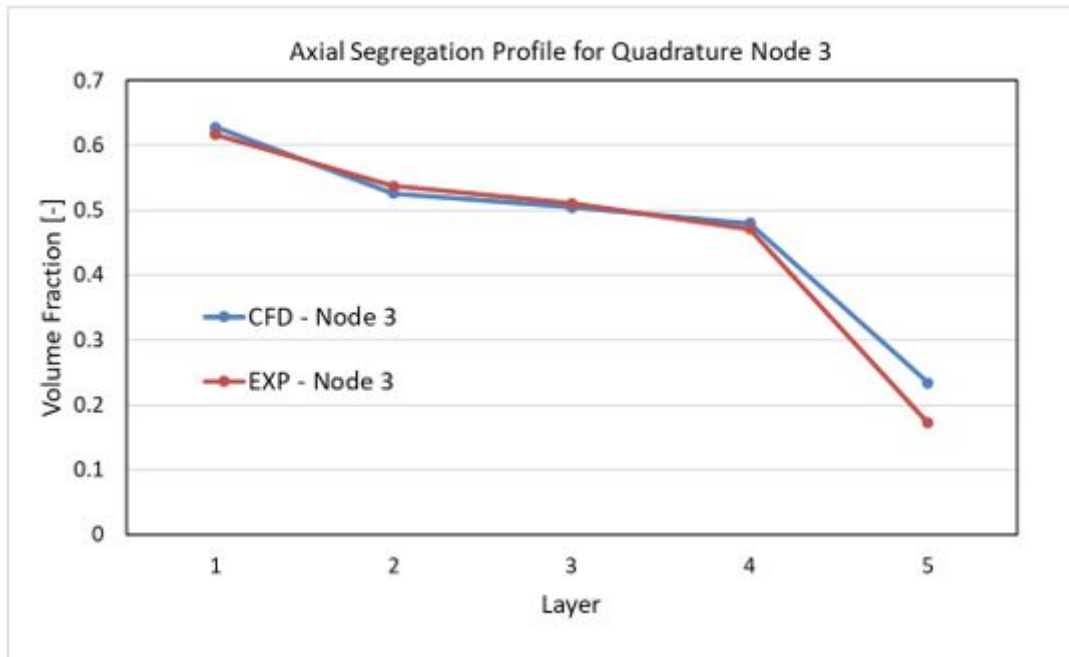


Figure 5.24: Axial Segregation Profile for Quadrature Node 3 at Pseudo-stationary conditions ($t = 10\text{s}$) for fluidization velocity = 0.10 m/s

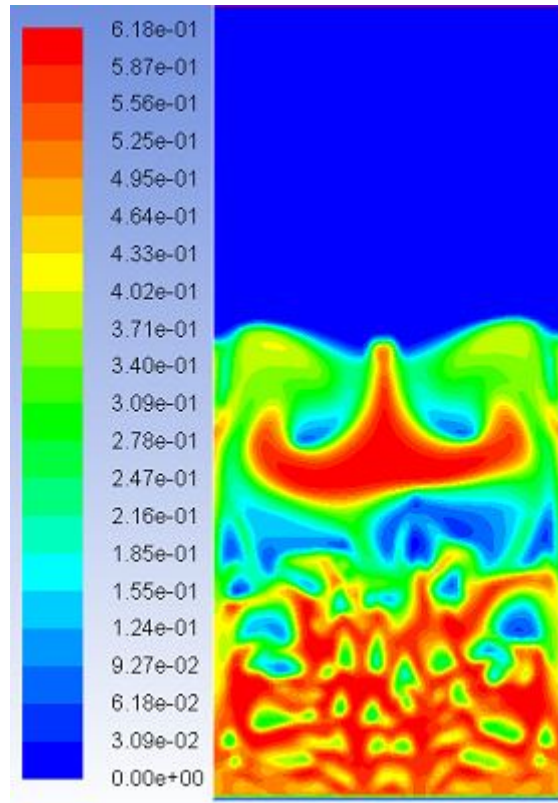


Figure 5.25: Contour of Volume fraction for Quadrature Node 3 at pseudo-stationary conditions ($t = 10s$) for fluidization velocity = 0.10 m/s

Similarly, quadrature node 1 being the smallest in size is expected to decrease in number as we move from top to bottom of the bed. This is exactly what we observe from Figure 5.26 which shows the axial profile of quadrature node 1 at pseudo-stationary conditions. We observe good agreement with experiment except for the bottom of the bed where quadrature node 1 is over-predicted by approximately about 30%.

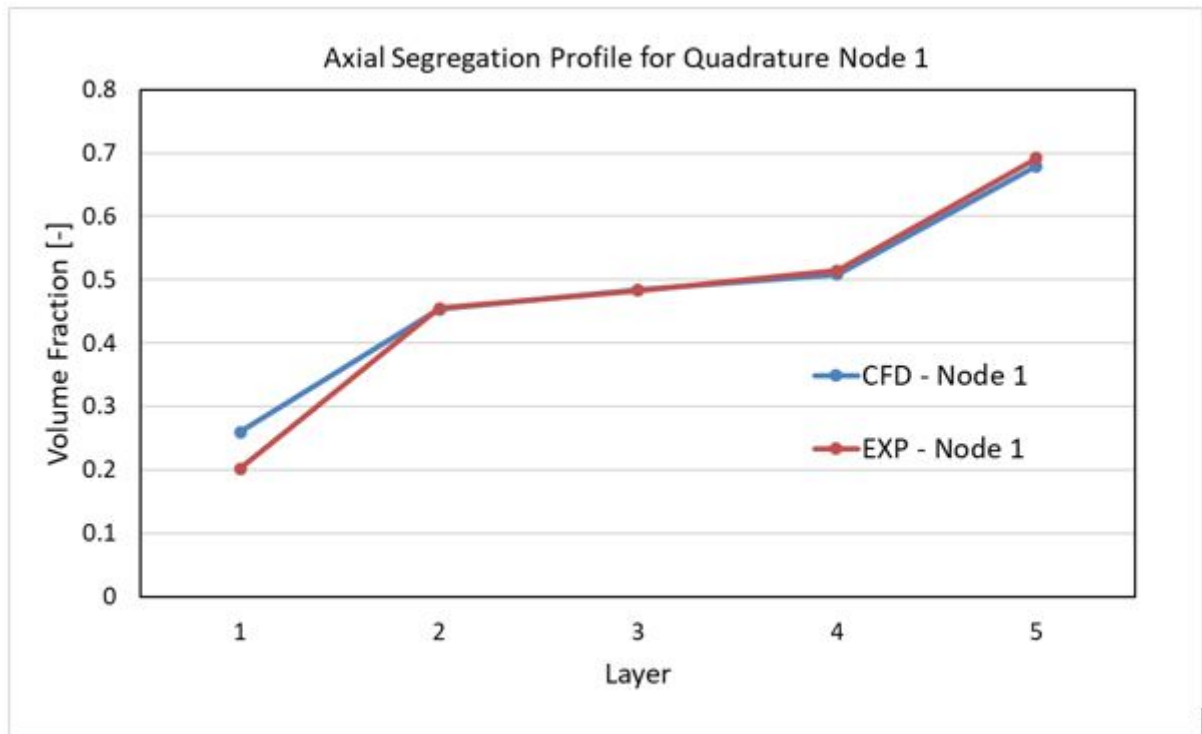


Figure 5.26: Axial Segregation Profile for Quadrature Node 1 at Pseudo-stationary conditions ($t = 10s$) for fluidization velocity = 0.10 m/s

The above results were derived from the implementation of QMOM model with moments being transported with different velocities. From the literature, QMOM model with the quadrature classes advected by the same velocity field such that the particles share the same velocity will not be suitable for simulating segregation.

Chapter 6

Conclusions and Future Work

Fluidization is used in several industrial processes, for example in waste disposal, food processing, pharmaceutical applications, energy conversions and so on. The task of designing fluidized bed reactors is prodigious as a result of numerous considerations which significantly affects the suspension fluid dynamics and in turn the product. Some of these considerations include but are not limited to fluid and particle properties, process variables, geometry change, aggregation, growth, breakage e.t.c. With the advances recorded in computational softwares and processes, CFD has become a very useful tool for researchers and design engineers alike in modelling fluidized bed reactors. However, the numerical simulations are as good as the mathematical models that the code solves. In modelling, these three aspects avail: the mathematical models, their implementation and the level of accuracy at which they predict real systems. The simulations have to be accurate enough to provide reliable information.

In this work, we have recognised that designing multiphase polydisperse systems are still subject to great uncertainties; reason being that the multiphase polydisperse systems (in our case fluidized beds) involve many physical and chemical phenomena occurring simultaneously and are hard to model. In chapter one, we presented a brief introduction to the theoretical concept and computational modelling of fluidized systems. This chapter was concluded with an introduction of the motivations and the objectives we seek to achieve at the completion of the project. In chapter two, we carried out a brief literature review on past work. We considered works involving gas-solid fluidized bed systems, hybrid CFD-PBM models and quadrature based method of moments. We then proceeded to present the generalized population balance equation (GPBE) which governs the distribution of the population of particles over any property of interest. The GPBE captures the physics of the continuous changes of the PSD in time and space associated with our system such that no additional conservation equations are needed. But its dimensionality is higher than that of classical transport equations. We therefore replace the GPBE with a set of three-dimensional averaged transport equations that govern the moments of the distribution using the method of moments but the set is often unclosed and must be made self-sufficient. We presented two methods to overcome the problem: the direct quadrature method of moments (DQMOM) and the quadrature method of moments (QMOM). Both represent the population by a finite number of classes, each with a number density and a specific set of internal properties; we therefore have a population distribution

approximated by a quadrature formula that turns integrals into summations posing a solution to the problem of closure. Whereas QMOM tracks the moments of the distribution and back-calculates the quadrature nodes and weights, the DQMOM on the other hand directly tracks the nodes and the weights. We described the working principle of the DQMOM model to better understand how, in principle, it is similar to the QMOM model. This was the basis for comparing the results from our work with those of Fan et al. (2004) derived from DQMOM implemented in MFIx.

One of the success factors for modelling a dense polydisperse system with a changing PSD lies in choosing appropriate aggregation and breakage functions that correctly depict the system. In chapter three, we presented aggregation and breakage kernels from the kinetic theory of granular flow (KTGF) to describe the changes in size resulting from breakage and aggregation in the disperse phase and incorporated same in the PBE. These kernels which are effective parameters in the PBE consist of two parts: the collision frequency which informs us of the number of particle pairs available for collision or the number of collisions undergone by each particle per time and a second part that tells us the success of aggregation and breakage resulting from these collisions. These kernels are dependent on a number of operating conditions and various process parameters. So developing apriori expressions from a theoretical stand-point is a difficult task. In chapter 4, we presented a brief description of the modelling equations implemented in Fluent and MFIx (for the work of Fan et al. (2004))

In chapter 5, while presenting simulation results, we reported the problem of moment corruption especially for simulations using higher order upwind scheme and those tracking more than six NDF moments. When the CFD is coupled with the PBE through the solution of moment transport equations, the calculated moments have a tendency to form invalid sets because in QMOM, the moment transport equations are solved independently. Hence, the resulting moments obtained via the inversion procedure might not preserve the relationships between the moments which make them part of a valid set. A valid moment set provides crucial information about a physical distribution, whereas an invalid sequence does not represent any valid physical distribution. Vikas et al. (2010) specifically proposed the so-called “quasi-high-order” advection scheme for the QMOM to preserve realizability of moment sets. However, owing to the fact the QMOM model used in this work is built on an already existing numerical solver, modifying the available advection scheme is either impossible or not desired (Afzalifar et al., 2017). Therefore, we considered and also implemented alternative solutions as seen in the literature.

One of the solutions reported in the literature is the McGraw (2006) correction method referred to as the *filter method* (McGraw, 2012). Because moment corruption is likely as a result of improper assignment of one, or at most a few of the moments in the sequence, we adjusted only those moments. For a sufficiently long sequence the filter method is the way to achieve this goal. In this method a difference table (see Appendix D.1.1) is constructed and used to identify the order of the moment that is corrupted and then provide an optimal correction by minimizing the sum of the squared differences of the second order differences, which through correction ensures the smoothness of $\ln(m_k)$ sequence. This correction method by McGraw (2006) only takes account of the convexity condition,

which is only necessary but not a sufficient condition. The convexity condition is satisfied if and only if all the elements of the second-order difference vector are non-negative. In some cases, the convexity test is passed but the moment set is still unphysical. In this case we will need to subject the moments to a test that identifies both necessary and sufficient conditions such as the positivity of the Hankel-Hadamard determinants (see Appendix D.1.2) constructed from the moments.

McGraw (2012) also proposed correcting invalid moment sets using the positive alpha sequence enforcement (PASE) method. This method first subjects the moment set to a check to ascertain its validity. In this case, in place of the Hankel-Hadamard determinants, another set of non-negative quantities, the alpha sequence, investigated by Gordon and generated by him using the P-D algorithm (Gordon, 1968) is used to investigate the validity of the moment set. Inspection of the alpha sequence will indicate immediately whether or not a given moment sequence is valid and provide a recipe for correction if the tested sequence proves to be invalid. To obtain the alpha sequence, the widely available Numerical Recipes subroutine ORTHOG is used to first obtain a tridiagonal Jacobi matrix from the moment sequence. The method of PASE correction is to set the first negative entry and all higher values of the alpha sequence to zero. The modified alpha sequence generates a valid sequence. The quadrature abscissas and weights are obtained by solving the eigenvalue problem associated with the Jacobi matrix.

Wright (2007) replaced nonrealizable moment set using a log-normal distribution or the arithmetic average of two log-normal distributions by retaining the zeroth and third moments from the original moment sequence. For a log-normal distribution, a generic moment can be written as:

$$m_k = N_T \exp \left(k\mu + \frac{k^2\sigma^2}{2} \right) \quad (6.1)$$

where N_T is the number density, μ and σ correspond to the mean and variance of the PSD.

We employed a systematic blend of these methods to achieve correction of corrupted moments. However, we observed that the correction algorithm was only effective when the quadrature classes were transported in the same velocity field compared to when they had different velocities.

We further probed the robustness of the model by checking its ability to predict segregation of fluidized powders. The results obtained were in good agreement with experiment.

6.1 Future Work

1. **Experimental Investigations** : In chapter 3 and 4, we presented the aggregation and breakage kernels. The equations 3.64, 3.70 and 3.76 represent the aggregation kernel, breakage kernel and daughter distribution function respectively and they were implemented in the different CFD codes employed in our work. In order to accurately predict the aggregation and breakage contributions, we need to be able

to determine the values of Ψ_a and Ψ_b . One way to do this is to run an experiment to achieve steady-state in which the system is fully mixed and the NDF is the same at every point in space. Afterwards, a CFD simulation with guessed values of Ψ_a and Ψ_b is initiated and the results compared with experiment to know if the NDF predicted is similar to that obtained from experiment. If the guessed values fail to predict accurately the system under investigation, another simulation is run with new values of Ψ_a and Ψ_b until we are able to obtain values of the aggregation and breakage efficiencies which mimic the NDF obtained from experiment. This method is prohibitive because the CFD simulations take long to run and we might also have to run several simulations before we achieve correct estimated values for Ψ_a and Ψ_b which gives a decent estimate of the system under investigation. In view of this, one future work to anticipate is a scientific methodology to approximate values of Ψ_a and Ψ_b that is at the same time not time demanding. In this case the experimental conditions can be modified to determine their influence on the PSD and the aggregation and breakage efficiencies.

2. ***Modelling other vessel configurations using quadrature based methods*** : In mathematical modelling, there is the need for the model to be robust. In other words, when variables or assumptions are altered the model is able to give correct predictions without failure. Altering the geometrical considerations of the vessel: cylindrical to tapered and finally to spout beds is another work that can be done to test the robustness of these models. Wormsbecker et al. (2009) studied the influence of vessel geometry on the hydrodynamic behaviour and final quality of the product of cylindrical and conical fluidized bed dryers by maintaining the same drying rates. Proportional volumetric rates of the fluid phase per unit mass of granule dried were maintained for both vessel configurations. However, results showed that while the cylindrical bed defluidized at the bottom due to cohesive granules forming aggregates, the conical bed generated a spout-like circulation pattern even while bubbling was maintained at the core of the bed. This is a work that can be independently pursued by researchers in the particle technology field.

Appendices

Appendix A

A.1 Population Balance Equation for Well-Mixed Systems

Most engineering systems involving dispersed-phase are mostly treated as well-mixed. In such cases, we are not interested in the spatial distribution of particles but rather in knowing how the particles are distributed over the internal space coordinates (e.g size space in our case). We shall refer to Equation 2.46:

$$\partial_t f_n(\mathbf{x}, s, t) + \nabla_{\mathbf{x}} \cdot (f_n(\mathbf{x}, s, t) \langle \mathbf{v} | s \rangle) = h_n(\mathbf{x}, s, t) \quad (2)$$

which is distributed in external phase space and then we shall average over the external phase space to transform Eq.2 to a more useful equation for mixed systems.

Let us consider a differential volume $d\mathbf{x}$ in a suitable finite volume $\Lambda_{\mathbf{x}}(t)$ in external phase space and then integrate Eq.2 over this region. This yields:

$$\int_{\Lambda_{\mathbf{x}}} [\partial_t f_n(\mathbf{x}, s, t) + \nabla_{\mathbf{x}} \cdot (f_n(\mathbf{x}, s, t) \langle \mathbf{v} | s \rangle) - h_n(\mathbf{x}, s, t)] d\mathbf{x} = 0 \quad (3)$$

Since the suspension is mixed, f_n and h_n are functions only of time and size and can be taken out of the integration. Therefore the first term in Eq. 3 may be written as:

$$\partial_t f_n(s, t) \int_{\Lambda_{\mathbf{x}}} d\mathbf{x} = \partial_t f_n(s, t) \Lambda_{\mathbf{x}} \quad (4)$$

Similarly, the third term in Eq. 3 may be written as:

$$h_n(s, t) \int_{\Lambda_{\mathbf{x}}} d\mathbf{x} = h_n(s, t) \Lambda_{\mathbf{x}} \quad (5)$$

The second term in Eq.3 can be expressed as a surface integral of the population flux (Gauss theorem):

$$\int_{\Lambda_{\mathbf{x}}} \nabla_{\mathbf{x}} \cdot (f_n \langle \mathbf{v} | s \rangle) d\mathbf{x} = \int_{\partial \Lambda_{\mathbf{x}}} f_n \langle \mathbf{v} | s \rangle \cdot d\mathbf{S}_{\mathbf{x}} \quad (6)$$

where $d\mathbf{S}_{\mathbf{x}}$ and $\langle \mathbf{v} | s \rangle$ are the differential surface vector and the size-dependent mean particle velocity normal to $\partial \Lambda_{\mathbf{x}}$, the surface bounding the control volume $\Lambda_{\mathbf{x}}(t)$. The volume can be considered to have arbitrary number of input and output streams of flow rate $\mathbf{v}_k \cdot d\mathbf{S}_k = Q_k$ and population density $f_k(s, t)$, assumed to be independent of position .

Thus, we can represent this term as the integral of $f_k \mathbf{v}_k \cdot d\mathbf{S}_k$ of all the individual streams k in and out of the system via the system boundary as:

$$\int_{\partial\Lambda_x} f_k \mathbf{v}_k \cdot d\mathbf{S}_k = \sum_k Q_k f_k \quad (7)$$

where Q_k is negative for flow into the volume and positive for flow out of the volume (sign convention assumes that outward-directed normal to the surface is positive).

Also we can consider a change in volume as a result of accumulation at the free interface. If we integrate over the entire control volume, we can represent the contribution of this term as:

$$f_n(s,t) \int_{\Lambda_x} \partial_t d\mathbf{x} = f_n(s,t) \partial_t \int_{\Lambda_x} d\mathbf{x} = f_n(s,t) \partial_t \Lambda_x \quad (8)$$

Hence, the total contribution of the spatial population flux divergence term is given as:

$$\int_{\partial\Lambda_x} f_n \langle \mathbf{v}|s \rangle \cdot d\mathbf{S}_x = \sum_k Q_k f_k + f_n(s,t) \partial_t \Lambda_x \quad (9)$$

and the spatial-averaged population balance equation may be written as:

$$\partial_t f_n(s,t) \Lambda_x + f_n(s,t) \partial_t \Lambda_x = h_n(s,t) \Lambda_x - \sum_k Q_k f_k \quad (10)$$

For a constant-volume system with no input and output streams, Eq.10 reduces to:

$$\partial_t f_n(s,t) = h_n(s,t) \quad (11)$$

Appendix B

B.1 Derivation of the DQMOM Multi-Fluid Model Equations

In order to remain consistent with the variables used in the multi-fluid model, the weight and nodes will be related to the volume fraction ϕ_{si} and weighted node $\phi_{si}s_i$ of each of the solid phase. The volume fraction and the weighted node of the i -th solid phase can be related to the weights and nodes as follows:

$$\phi_{si} = k_v s_i^3 n_i \quad (12)$$

and

$$\phi_{si}s_i = k_v s_i^4 n_i \quad (13)$$

where k_v is the volumetric shape factor which equals $\pi/6$ for spherical particles, s_i and n_i represent the nodes and weights respectively.

B.1.1 Transport Equation for the Volume Fraction ϕ_{si}

If we substitute Eq.12 into the transport equation for the volume fraction ϕ_{si} we get:

$$\begin{aligned} \partial_t(\phi_{si}\rho_{si}) + \nabla_{\mathbf{x}} \cdot (\phi_{si}\rho_{si}\mathbf{u}_{si}) &= \partial_t(k_v s_i^3 n_i \rho_{si}) + \nabla_{\mathbf{x}} \cdot (k_v s_i^3 n_i \rho_{si} \mathbf{u}_{si}) \\ &= k_v \rho_{si} [\partial_t(s_i^3 n_i) + \nabla_{\mathbf{x}} \cdot (s_i^3 n_i \mathbf{u}_{si})] \end{aligned} \quad (14)$$

where the density of the solid particles ρ_{si} is treated as a constant. We can expand the first term in the bracket in Eq.14 thus:

$$\begin{aligned} \partial_t(s_i^3 n_i) &= n_i \partial_t s_i^3 + s_i^3 \partial_t n_i \\ &= 3s_i^2 n_i \partial_t s_i + s_i^3 \partial_t n_i \\ &= 3s_i^2 [\partial_t(n_i s_i) - s_i \partial_t n_i] + s_i^3 \partial_t n_i \\ &= 3s_i^2 \partial_t(n_i s_i) - 3s_i^3 s_i \partial_t n_i + s_i^3 \partial_t n_i \\ &= 3s_i^2 \partial_t(n_i s_i) - 2s_i^3 \partial_t n_i \end{aligned} \quad (15)$$

and the second term in the bracket in Eq.14 is expanded thus:

$$\begin{aligned}
\nabla_{\mathbf{x}} \cdot (s_i^3 n_i \mathbf{u}_{si}) &= \mathbf{u}_{si} \nabla_{\mathbf{x}} (s_i^3 n_i) + s_i^3 n_i \nabla_{\mathbf{x}} \cdot \mathbf{u}_{si} \\
&= \mathbf{u}_{si} (s_i^3 \nabla_{\mathbf{x}} n_i + n_i \nabla_{\mathbf{x}} s_i^3) + s_i^3 n_i \nabla_{\mathbf{x}} \cdot \mathbf{u}_{si} \\
&= \mathbf{u}_{si} (s_i^3 \nabla_{\mathbf{x}} n_i + 3s_i^2 n_i \nabla_{\mathbf{x}} s_i) + s_i^3 n_i \nabla_{\mathbf{x}} \cdot \mathbf{u}_{si} \\
&= \mathbf{u}_{si} [s_i^3 \nabla_{\mathbf{x}} n_i + 3s_i^2 (\nabla_{\mathbf{x}} (n_i s_i) - s_i \nabla_{\mathbf{x}} n_i)] + s_i^3 n_i \nabla_{\mathbf{x}} \cdot \mathbf{u}_{si} \\
&= s_i^3 \mathbf{u}_{si} \nabla_{\mathbf{x}} n_i + 3s_i^2 \mathbf{u}_{si} \nabla_{\mathbf{x}} (n_i s_i) - 3s_i^3 \mathbf{u}_{si} \nabla_{\mathbf{x}} n_i + s_i^3 n_i \nabla_{\mathbf{x}} \cdot \mathbf{u}_{si} \\
&= 3s_i^2 \mathbf{u}_{si} \nabla_{\mathbf{x}} (n_i s_i) - 2s_i^3 \mathbf{u}_{si} \nabla_{\mathbf{x}} n_i + s_i^3 n_i \nabla_{\mathbf{x}} \cdot \mathbf{u}_{si}
\end{aligned} \tag{16}$$

Substituting Eqs.15 and 16 in to Eq.14 we have:

$$\begin{aligned}
&\partial_t (\phi_{si} \rho_{si}) + \nabla_{\mathbf{x}} \cdot (\phi_{si} \rho_{si} \mathbf{u}_{si}) \\
&= \mathbf{u}_{si} (s_i^4 \nabla_{\mathbf{x}} n_i + n_i \nabla_{\mathbf{x}} s_i^4) + s_i^4 n_i \nabla_{\mathbf{x}} \cdot \mathbf{u}_{si} \\
&= \mathbf{u}_{si} (s_i^4 \nabla_{\mathbf{x}} n_i + 4s_i^3 n_i \nabla_{\mathbf{x}} s_i) + s_i^4 n_i \nabla_{\mathbf{x}} \cdot \mathbf{u}_{si} \\
&= \mathbf{u}_{si} [s_i^4 \nabla_{\mathbf{x}} n_i + 4s_i^3 (\nabla_{\mathbf{x}} (n_i s_i) - s_i \nabla_{\mathbf{x}} n_i)] + s_i^4 n_i \nabla_{\mathbf{x}} \cdot \mathbf{u}_{si} \\
&= s_i^4 \mathbf{u}_{si} \nabla_{\mathbf{x}} n_i + 4s_i^3 \mathbf{u}_{si} \nabla_{\mathbf{x}} (n_i s_i) - 4s_i^4 \mathbf{u}_{si} \nabla_{\mathbf{x}} n_i + s_i^4 n_i \nabla_{\mathbf{x}} \cdot \mathbf{u}_{si} \\
&= 4s_i^3 \mathbf{u}_{si} \nabla_{\mathbf{x}} (n_i s_i) - 3s_i^4 \mathbf{u}_{si} \nabla_{\mathbf{x}} n_i + s_i^4 n_i \nabla_{\mathbf{x}} \cdot \mathbf{u}_{si}
\end{aligned} \tag{17}$$

Substituting Eqs.28 and 22 in to Eq.25 we have:

$$\begin{aligned}
&\partial_t (\phi_{si} s_i \rho_{si}) + \nabla_{\mathbf{x}} \cdot (\phi_{si} s_i \rho_{si} \mathbf{u}_{si}) \\
&= k_v \rho_{si} [3s_i^2 \partial_t (n_i s_i) - 2s_i^3 \partial_t n_i + 3s_i^2 \mathbf{u}_{si} \nabla_{\mathbf{x}} (n_i s_i) - 2s_i^3 \mathbf{u}_{si} \nabla_{\mathbf{x}} n_i + s_i^3 n_i \nabla_{\mathbf{x}} \cdot \mathbf{u}_{si}] \\
&= k_v \rho_{si} [3s_i^2 \partial_t (n_i s_i) - 2s_i^3 \partial_t n_i + 3s_i^2 \mathbf{u}_{si} \nabla_{\mathbf{x}} (n_i s_i) - 2s_i^3 \mathbf{u}_{si} \nabla_{\mathbf{x}} n_i + 3s_i^3 n_i \nabla_{\mathbf{x}} \cdot \mathbf{u}_{si} - 2s_i^3 n_i \nabla_{\mathbf{x}} \cdot \mathbf{u}_{si}] \\
&= k_v \rho_{si} [3s_i^2 \partial_t (n_i s_i) + 3s_i^2 \mathbf{u}_{si} \nabla_{\mathbf{x}} (n_i s_i) + 3s_i^3 n_i \nabla_{\mathbf{x}} \cdot \mathbf{u}_{si} - 2s_i^3 \partial_t n_i - 2s_i^3 \mathbf{u}_{si} \nabla_{\mathbf{x}} n_i - 2s_i^3 n_i \nabla_{\mathbf{x}} \cdot \mathbf{u}_{si}] \\
&= 3s_i^2 k_v \rho_{si} [\partial_t (n_i s_i) + \mathbf{u}_{si} \nabla_{\mathbf{x}} (n_i s_i) + s_i n_i \nabla_{\mathbf{x}} \cdot \mathbf{u}_{si}] - 2s_i^3 k_v \rho_{si} (\partial_t n_i + \mathbf{u}_{si} \nabla_{\mathbf{x}} n_i + n_i \nabla_{\mathbf{x}} \cdot \mathbf{u}_{si}) \\
&= 3s_i^2 k_v \rho_{si} c_i^s - 2s_i^3 k_v \rho_{si} c_i^n
\end{aligned} \tag{18}$$

where by definition,

$$c_i^s = \partial_t (n_i s_i) + \mathbf{u}_{si} \nabla_{\mathbf{x}} (n_i s_i) + s_i n_i \nabla_{\mathbf{x}} \cdot \mathbf{u}_{si} \quad ; \quad c_i^n = \partial_t n_i + \mathbf{u}_{si} \nabla_{\mathbf{x}} n_i + n_i \nabla_{\mathbf{x}} \cdot \mathbf{u}_{si} \tag{19}$$

B.1.2 Transport Equation for the Weighted Node $\phi_{si} s_i$

If we substitute Eq.13 into the transport equation for the the weighted node $\phi_{si} s_i$ we get:

$$\begin{aligned}
\partial_t (\phi_{si} s_i \rho_{si}) + \nabla_{\mathbf{x}} \cdot (\phi_{si} s_i \rho_{si} \mathbf{u}_{si}) &= \partial_t (k_v s_i^4 n_i \rho_{si}) + \nabla_{\mathbf{x}} \cdot (k_v s_i^4 n_i \rho_{si} \mathbf{u}_{si}) \\
&= k_v \rho_{si} [\partial_t (s_i^4 n_i) + \nabla_{\mathbf{x}} \cdot (s_i^4 n_i \mathbf{u}_{si})]
\end{aligned} \tag{20}$$

where the density of the solid particles ρ_{si} is a constant. We can expand the first term in the bracket in Eq.25 thus:

$$\begin{aligned}
\partial_t(s_i^4 n_i) &= n_i \partial_t s_i^4 + s_i^4 \partial_t n_i \\
&= 4s_i^3 n_i \partial_t s_i + s_i^4 \partial_t n_i \\
&= 4s_i^3 [\partial_t(n_i s_i) - s_i \partial_t n_i] + s_i^4 \partial_t n_i \\
&= 4s_i^3 \partial_t(n_i s_i) - 4s_i^4 s_i \partial_t n_i + s_i^4 \partial_t n_i \\
&= 4s_i^3 \partial_t(n_i s_i) - 3s_i^4 \partial_t n_i
\end{aligned} \tag{21}$$

and the second term in the bracket in Eq.25 is expanded thus:

$$\begin{aligned}
\nabla_{\mathbf{x}} \cdot (s_i^4 n_i \mathbf{u}_{si}) &= \mathbf{u}_{si} \nabla_{\mathbf{x}} (s_i^4 n_i) + s_i^4 n_i \nabla_{\mathbf{x}} \cdot \mathbf{u}_{si} \\
&= \mathbf{u}_{si} (s_i^4 \nabla_{\mathbf{x}} n_i + n_i \nabla_{\mathbf{x}} s_i^4) + s_i^4 n_i \nabla_{\mathbf{x}} \cdot \mathbf{u}_{si} \\
&= \mathbf{u}_{si} (s_i^4 \nabla_{\mathbf{x}} n_i + 4s_i^3 n_i \nabla_{\mathbf{x}} s_i) + s_i^4 n_i \nabla_{\mathbf{x}} \cdot \mathbf{u}_{si} \\
&= \mathbf{u}_{si} [s_i^4 \nabla_{\mathbf{x}} n_i + 4s_i^3 (\nabla_{\mathbf{x}} (n_i s_i) - s_i \nabla_{\mathbf{x}} n_i)] + s_i^4 n_i \nabla_{\mathbf{x}} \cdot \mathbf{u}_{si} \\
&= s_i^4 \mathbf{u}_{si} \nabla_{\mathbf{x}} n_i + 4s_i^3 \mathbf{u}_{si} \nabla_{\mathbf{x}} (n_i s_i) - 4s_i^4 \mathbf{u}_{si} \nabla_{\mathbf{x}} n_i + s_i^4 n_i \nabla_{\mathbf{x}} \cdot \mathbf{u}_{si} \\
&= 4s_i^3 \mathbf{u}_{si} \nabla_{\mathbf{x}} (n_i s_i) - 3s_i^4 \mathbf{u}_{si} \nabla_{\mathbf{x}} n_i + s_i^4 n_i \nabla_{\mathbf{x}} \cdot \mathbf{u}_{si}
\end{aligned} \tag{22}$$

Substituting Eqs.28 and 22 in to Eq.25 we have:

$$\begin{aligned}
&\partial_t (\phi_{si} s_i \rho_{si}) + \nabla_{\mathbf{x}} \cdot (\phi_{si} s_i \rho_{si} \mathbf{u}_{si}) \\
&= k_v \rho_{si} [4s_i^3 \partial_t (n_i s_i) - 3s_i^4 \partial_t n_i + 4s_i^3 \mathbf{u}_{si} \nabla_{\mathbf{x}} (n_i s_i) - 3s_i^4 \mathbf{u}_{si} \nabla_{\mathbf{x}} n_i + s_i^4 n_i \nabla_{\mathbf{x}} \cdot \mathbf{u}_{si}] \\
&= k_v \rho_{si} [4s_i^3 \partial_t (n_i s_i) - 3s_i^2 \partial_t n_i + 4s_i^3 \mathbf{u}_{si} \nabla_{\mathbf{x}} (n_i s_i) - 3s_i^4 \mathbf{u}_{si} \nabla_{\mathbf{x}} n_i + 4s_i^4 n_i \nabla_{\mathbf{x}} \cdot \mathbf{u}_{si} - 3s_i^4 n_i \nabla_{\mathbf{x}} \cdot \mathbf{u}_{si}] \\
&= k_v \rho_{si} [4s_i^3 \partial_t (n_i s_i) + 4s_i^3 \mathbf{u}_{si} \nabla_{\mathbf{x}} (n_i s_i) + 4s_i^4 n_i \nabla_{\mathbf{x}} \cdot \mathbf{u}_{si} - 3s_i^4 \partial_t n_i - 3s_i^4 \mathbf{u}_{si} \nabla_{\mathbf{x}} n_i - 3s_i^4 n_i \nabla_{\mathbf{x}} \cdot \mathbf{u}_{si}] \\
&= 4s_i^3 k_v \rho_{si} [\partial_t (n_i s_i) + \mathbf{u}_{si} \nabla_{\mathbf{x}} (n_i s_i) + s_i n_i \nabla_{\mathbf{x}} \cdot \mathbf{u}_{si}] - 3s_i^4 k_v \rho_{si} (\partial_t n_i + \mathbf{u}_{si} \nabla_{\mathbf{x}} n_i + n_i \nabla_{\mathbf{x}} \cdot \mathbf{u}_{si}) \\
&= 4s_i^3 k_v \rho_{si} c_i^s - 3s_i^4 k_v \rho_{si} c_i^n
\end{aligned} \tag{23}$$

where by definition,

$$c_i^s = \partial_t (n_i s_i) + \mathbf{u}_{si} \nabla_{\mathbf{x}} (n_i s_i) + s_i n_i \nabla_{\mathbf{x}} \cdot \mathbf{u}_{si} \quad ; \quad c_i^n = \partial_t n_i + \mathbf{u}_{si} \nabla_{\mathbf{x}} n_i + n_i \nabla_{\mathbf{x}} \cdot \mathbf{u}_{si} \tag{24}$$

Appendix C

C.1 Calculation of Sauter-Mean Particle Size, Nodes and Volume Fractions from Results of Numerical Simulations

The numerical simulation results give solutions for the weights and nodes. We have already shown how we can relate the weight to the volume fraction so that we can associate each node with the volume fraction and describe the PSD using the nodes and the volume fractions (the modified weight). However, in our simulation results these values differ from cell to cell as expected and we therefore need to work with an average value. The average value on the other hand can be calculated in different ways. Hence, we have to select an average with physical significance to our system. The average values reported in this work for the sauter-mean particle size, the volume fractions and nodes are calculated as follows:

C.1.1 The Sauter-Mean Particle Size d_{32}

The sauter-mean particle size (also the volume -average mean particle size) d_{32} , is the ratio between the third moment m_3 and the second moment m_2 of the PSD. This may be written as:

$$d_{32} = \frac{m_3}{m_2} = \frac{\int_{\Lambda_s} s^3 f_n(s) ds}{\int_{\Lambda_s} s^2 f_n(s) ds} \quad (25)$$

To define $f_n(s)$, we consider a simple case where the domain is made up of two cells with volumes v_1 and v_2 . We can therefore write:

$$v_1 f_1(s) ds + v_2 f_2(s) ds = (v_1 + v_2) f_n(s) ds \quad (26)$$

which means that the total number of particles in the computational domain is conserved. Hence, we may write:

$$f_n(s) = \frac{v_1 f_1(s) + v_2 f_2(s)}{v_1 + v_2} \quad (27)$$

Generally,

$$f_n(s) = \frac{\sum_{i=1}^N v_i f_i(s)}{\sum_{i=1}^N v_i} \quad (28)$$

From the quadrature approximation,

$$f_i(s) = \sum_{k=1}^M n_{i,k} \delta(s - s_{i,k}) \quad (29)$$

Substituting Eqs. 28 and 29 into Eq. 25 we have

$$d_{32} = \frac{\int_{\Lambda_s} s^3 [\sum_{i=1}^N v_i f_i(s) / \sum_{i=1}^N v_i] ds}{\int_{\Lambda_s} s^2 [\sum_{i=1}^N v_i f_i(s) / \sum_{i=1}^N v_i] ds} = \frac{\int_{\Lambda_s} s^3 [\sum_{i=1}^N v_i f_i(s)] ds}{\int_{\Lambda_s} s^2 [\sum_{i=1}^N v_i f_i(s)] ds} = \frac{\sum_{i=1}^N v_i \int_{\Lambda_s} s^3 \sum_{k=1}^M n_{i,k} \delta(s - s_{i,k}) ds}{\sum_{i=1}^N v_i \int_{\Lambda_s} s^2 \sum_{k=1}^M n_{i,k} \delta(s - s_{i,k}) ds} \quad (30)$$

So that,

$$d_{32} = \frac{\sum_{i=1}^N v_i \sum_{k=1}^M n_{i,k} \int_{\Lambda_s} s^3 \delta(s - s_{i,k}) ds}{\sum_{i=1}^N v_i \sum_{k=1}^M n_{i,k} \int_{\Lambda_s} s^2 \delta(s - s_{i,k}) ds} = \frac{\sum_{i=1}^N v_i \sum_{k=1}^M n_{i,k} s_{i,k}^3}{\sum_{i=1}^N v_i \sum_{k=1}^M n_{i,k} s_{i,k}^2} \quad (31)$$

Hence,

$$d_{32} = \frac{\sum_{i=1}^N m_{3,i} v_i}{\sum_{i=1}^N m_{2,i} v_i} \quad (32)$$

where N refers to the number of cells in the computational domain and M is the number of particle size class.

C.1.2 The average node, $\langle s_i \rangle$ and average volume fraction, $\langle \phi_{si} \rangle$

These are the average values of the nodes and volume fractions of the entire distribution characterised by $f_n(s)$. We have already shown that

$$f_n(s) = \frac{\sum_{i=1}^N v_i f_i(s)}{\sum_{i=1}^N v_i}$$

Also we can write:

$$f_n(s) = \sum_{k=1}^M \langle n_k \rangle \delta(s - \langle s_k \rangle) \quad (33)$$

If we consider the size-based volume-averaged moment of order p , $\langle m_p \rangle$:

$$\langle m_p \rangle = \int_0^\infty s^p f_n(s) ds \quad (34)$$

Substituting the quadrature approximation of Eq. 33 in Eq. 34, we have:

$$\langle m_p \rangle = \int_0^\infty s^p \sum_{k=1}^M \langle n_k \rangle \delta(s - \langle s_k \rangle) ds = \sum_{k=1}^M \langle n_k \rangle \int_0^\infty s^p \delta(s - \langle s_k \rangle) ds \quad (35)$$

This becomes

$$\langle m_p \rangle = \sum_{k=1}^M \langle n_k \rangle \langle s_k \rangle^p \quad (36)$$

Note that $\langle m_p \rangle$ is calculated thus:

$$\langle m_p \rangle = \int_0^\infty s^p f_n(s) ds = \frac{1}{V_T} \int_0^\infty s^p \sum_{i=1}^N v_i f_i(s) ds \quad (37)$$

where

$$V_T = \sum_{i=1}^N v_i$$

v_i is the volume of the i -th cell in the computational domain and N the number of computational cells in the domain.

$$\begin{aligned} \langle m_p \rangle &= \frac{1}{V_T} \int_0^\infty s^p \sum_{i=1}^N v_i \sum_{k=1}^M n_{i,k} \delta(s - s_{i,k}) ds \\ &= \frac{1}{V_T} \sum_{i=1}^N v_i \sum_{k=1}^M n_{i,k} \int_0^\infty s^p \delta(s - s_{i,k}) ds \\ &= \frac{1}{V_T} \sum_{i=1}^N v_i \sum_{k=1}^M n_{i,k} s_{i,k}^p \equiv \frac{1}{V_T} \sum_{i=1}^N v_i m_{i,p} \end{aligned} \quad (38)$$

Equation 36 is a non-linear equation in $2M$ unknowns, $\langle n_k \rangle$ and $\langle s_k \rangle^p$. For any value of M , the equation has $2M$ unknowns which can be found by writing and solving $2M$ equations for $2M$ different values of p . If we choose the first $2M$ volume-averaged moments we can construct a real symmetric tridiagonal matrix whose eigenvalues and eigenvectors coincide with the average quadrature nodes and weights respectively. This is possible using the product-difference algorithm already introduced in section 2.5.1. The reader is also referred to Marchisio and Fox (2013) for details on the PD algorithm.

Hence, the average volume fraction of the node k , $\langle \phi_{sk} \rangle$ can be derived from the values of the average weights of the quadrature nodes by using the relation:

$$\langle \phi_{sk} \rangle = k_v \langle s_k \rangle^3 \langle n_{sk} \rangle$$

Appendix D

Description of the convexity checks for valid moment sets

D.1.1 Difference Tables

The difference table is useful in spotting isolated errors in an ordered set of data, which is what the nonrealizable moment set problem requires (Lanczos, 1988). The construction of the difference table is straightforward. Table D1 shows the difference table for a six-moment sequence. The first column identifies the order of the moment in the moment set. The second column is the natural logarithm of the moment, $\ln m_k$. The subsequent columns are the i -th order differences respectively. The third column is the first order difference which is the difference of the elements in column 2. Column 4 which contains the second-order differences are the first-order differences of elements of column 3 and so on.

Table D1. Difference table showing the i -th order differences (d_k) for (a) uncorrupted and (b) corrupted moment sets

k	$\ln m_k$	d1	d2	d3	d4	d5
(a)						
0	0	1	2	0	0	0
1	1	3	2	0	0	
2	4	5	2	0		
3	9	7	2			
4	16	9				
5	25					
k	$\ln m_k$	d1	d2	d3	d4	d5
(b)						
0	0	1	2	-3	12	-30
1	1	3	-1	9	-18	
2	4	2	8	-9		
3	6	10	-1			
4	16	9				
5	25					

A necessary but not sufficient condition for the moment set to be valid is that $\ln m_k$ is a convex function of index k (Feller, 1971) which is satisfied if and only if the second-order differences are non-negative. In Table D1 (a), $\ln m_k$ was assigned a quadratic function of k and we notice that the non-negative entries in for $d2$ indicates that we can reasonably expect a smooth function of the index k and $\ln m_k$. In other words, the moment set is valid. However in Table D1 (b), the moment sequence was modified to corrupt the third order moment. We can see from the negative entries in the $d2$ column that the modified moment set violates convexity and the error spreads across the difference table as we move on to higher order differences.

D.1.2 Positive Alpha Sequence Enforcement (PASE)

The Hankel-Hadamard determinants constructed from the moments identifies the necessary and sufficient conditions for a valid moment sequence.

$$\Delta_n = \begin{vmatrix} m_0 & m_1 & \cdots & m_n \\ m_1 & m_2 & \cdots & m_{n+1} \\ \cdots & \cdots & \cdots & \cdots \\ m_n & m_{n+1} & \cdots & m_{2n} \end{vmatrix} \geq 0$$

$$\Delta_n^1 = \begin{vmatrix} m_1 & m_2 & \cdots & m_{n+1} \\ m_2 & m_3 & \cdots & m_{n+2} \\ \cdots & \cdots & \cdots & \cdots \\ m_{n+1} & m_{n+2} & \cdots & m_{2n+1} \end{vmatrix} \geq 0$$

We can replace the determinants above by another set of non-negative quantities, referred to as the alpha sequence, investigated and generated by Gordon (1968) using the PD algorithm. The moment sequence derived after an advection step of a model simulation can be inspected for the presence of unphysical moments by inspecting the alpha sequence. If any entry in the alpha sequence is negative, this is proof positive that the moment set is invalid. We construct the alpha sequence from the Hankel-Hadamard determinants. First, we consider ordering the determinants as follows:

$$e = \{\Delta_0, \Delta_0^1, \Delta_1, \Delta_1^1, \Delta_2, \Delta_2^1, \cdots\}$$

By inspection, we notice that the k -th member of this sequence introduces the $(k-1)$ -th moment. The alpha sequence has a lot of useful properties which can be used in identifying and correcting the corrupted moment in the moment sequence. Where the alpha sequence is defined as:

$$\alpha = \{\alpha_1, \alpha_2, \alpha_3, \cdots\}$$

where we define α_n as follows:

$$\alpha_1 = m_0$$

$$\alpha_2 = \Delta_0^1 = \frac{m_1}{m_0}$$

$$\alpha_3 = \frac{\Delta_1}{\Delta_0^1} = \frac{m_2 m_0 - m_1^2}{m_1 m_0}$$

$$\alpha_4 = \frac{\Delta_1^1}{\Delta_1 \Delta_0^1} = \frac{e_4 e_1}{e_3 e_2} = \frac{m_0(m_1 m_3 - m_2^2)}{m_1(m_0 m_2 - m_1^2)}$$

when n is even and greater than 4

$$\alpha_n = \frac{e_n e_{n-3}}{e_{n-1} e_{n-2}} = \frac{\Delta_{(n/2)-1}^1 \Delta_{(n/2)-2}}{\Delta_{(n/2)-2}^1 \Delta_{(n/2)-1}}$$

but when n is odd and greater than 5

$$\alpha_n = \frac{e_n e_{n-3}}{e_{n-1} e_{n-2}} = \frac{\Delta_{(n-1)/2} \Delta_{(n-3)/2}}{\Delta_{(n-3)/2}^1 \Delta_{(n-5)/2}^1}$$

The condition for a valid moment sequence is satisfied if and only if:

$$\alpha_n \geq 0 \quad (n = 1, 2, \dots)$$

Bibliography

- Acher, T., Dems, P., Lenz, S., Gobert, C. and Polifke, W. (2013), Validation of a quadrature method of moments for polydisperse flow in bubble columns including polydispersity, breakup and coalescence, *in* '8th Int. Conf. on Multiphase Flows, ICMF'.
- Afzalifar, A., Turunen-Saaresti, T. and Grönman, A. (2017), 'Nonrealizability problem with quadrature method of moments in wet-steam flows and solution techniques', *Journal of Engineering for Gas Turbines and Power* **139**(1), 012602.
- Akbari, V., Borhani, T. N. G., Shamiri, A., Aramesh, R., Hussain, M. A. and Hamid, M. K. A. (2015), '2d cfd-pbm simulation of hydrodynamic and particle growth in an industrial gas phase fluidized bed polymerization reactor', *Chemical Engineering Research and Design* **104**, 53–67.
- Arfken, G. and Weber, H. J. (1985), 'Mathematical methods for physicists academic', *New York* **1985**, 309.
- Buffo, A. and Marchisio, D. L. (2014), 'Modeling and simulation of turbulent polydisperse gas-liquid systems via the generalized population balance equation', *Reviews in Chemical Engineering* **30**(1), 73–126.
- Cammarata, L., Lettieri, P., Micale, G. D. and Colman, D. (2003), '2d and 3d cfd simulations of bubbling fluidized beds using eulerian-eulerian models', *International Journal of Chemical Reactor Engineering* **1**(1).
- Chapman, S. and Cowling, T. G. (1970), *The mathematical theory of non-uniform gases: an account of the kinetic theory of viscosity, thermal conduction and diffusion in gases*, Cambridge university press.
- Che, Y., Tian, Z., Liu, Z., Zhang, R., Gao, Y., Zou, E., Wang, S. and Liu, B. (2015), 'A cfd-pbm model considering ethylene polymerization for the flow behaviors and particle size distribution of polyethylene in a pilot-plant fluidized bed reactor', *Powder Technology* **286**, 107–123.
- Cooper, S. and Coronella, C. J. (2005), 'Cfd simulations of particle mixing in a binary fluidized bed', *Powder Technology* **151**(1-3), 27–36.
- Davidson, D. (2003), The role of computational fluid dynamics in process industries, *in* 'Frontiers of Engineering:: Reports on Leading-Edge Engineering from the 2002 NAE Symposium on Frontiers of Engineering', National Academies Press, p. 21.

- Di Renzo, A. and Di Maio, F. P. (2007), 'Homogeneous and bubbling fluidization regimes in dem-cfd simulations: hydrodynamic stability of gas and liquid fluidized beds', *Chemical Engineering Science* **62**(1), 116–130.
- Diemer, R. and Olson, J. (2002), 'A moment methodology for coagulation and breakage problems: Part 2—moment models and distribution reconstruction', *Chemical Engineering Science* **57**(12), 2211–2228.
- Ding, J. and Gidaspow, D. (1990), 'A bubbling fluidization model using kinetic theory of granular flow', *AIChE journal* **36**(4), 523–538.
- Fan, R., Marchisio, D. L. and Fox, R. O. (2004), 'Application of the direct quadrature method of moments to polydisperse gas–solid fluidized beds', *Powder technology* **139**(1), 7–20.
- Feller, W. (1971), 'Introduction to the theory of probability and its applications, vol. 2', *II (2. Ed.) New York: Wiley*.
- Fluent, A. (2009), '12.0 udf manual', *Ansys Inc*.
- Formisani, B., De Cristofaro, G. and Girimonte, R. (2001), 'A fundamental approach to the phenomenology of fluidization of size segregating binary mixtures of solids', *Chemical Engineering Science* **56**(1), 109–119.
- Gantt, J. A., Cameron, I. T., Litster, J. D. and Gatzke, E. P. (2006), 'Determination of coalescence kernels for high-shear granulation using dem simulations', *Powder Technology* **170**(2), 53–63.
- Geldart, D. (1973), 'Types of gas fluidization', *Powder technology* **7**(5), 285–292.
- Gelderbloom, S. J., Gidaspow, D. and Lyczkowski, R. W. (2003), 'Cfd simulations of bubbling/collapsing fluidized beds for three geldart groups', *AIChE journal* **49**(4), 844–858.
- Gera, D., Syamlal, M. and O'Brien, T. J. (2004), 'Hydrodynamics of particle segregation in fluidized beds', *International Journal of Multiphase Flow* **30**(4), 419–428.
- Gibilaro, L. (2001), *Fluidization dynamics*, Butterworth-Heinemann.
- Gidaspow, D. (1994), *Multiphase flow and fluidization: continuum and kinetic theory descriptions*, Academic press.
- Gidaspow, D., Bezburuah, R. and Ding, J. (1991), Hydrodynamics of circulating fluidized beds: kinetic theory approach, Technical report, Illinois Inst. of Tech., Chicago, IL (United States). Dept. of Chemical Engineering.
- Goldschmidt, M. J. V. (2001), *Hydrodynamic modelling of fluidised bed spray granulation*, Twente University Press.
- Gordon, R. G. (1968), 'Error bounds in equilibrium statistical mechanics', *Journal of Mathematical Physics* **9**(5), 655–663.

- Gu, Z., Su, J., Jiao, J. and Xu, X. Y. (2009), 'Simulation of micro-behaviors including nucleation, growth, and aggregation in particle system', *Science in China Series B: Chemistry* **52**(2), 241–248.
- Howley, M. A. and Glasser, B. J. (2002), 'Hydrodynamics of a uniform liquid-fluidized bed containing a binary mixture of particles', *Chemical engineering science* **57**(19), 4209–4226.
- Hrenya, C. M. and Sinclair, J. L. (1997), 'Effects of particle-phase turbulence in gas-solid flows', *AIChE Journal* **43**(4), 853–869.
- Huilin, L., Yunhua, Z., Ding, J., Gidaspow, D. and Wei, L. (2007), 'Investigation of mixing/segregation of mixture particles in gas–solid fluidized beds', *Chemical engineering science* **62**(1), 301–317.
- Huilin, L., Yurong, H., Gidaspow, D., Lidan, Y. and Yukun, Q. (2003), 'Size segregation of binary mixture of solids in bubbling fluidized beds', *Powder Technology* **134**(1), 86–97.
- Hulburt, H. M. and Katz, S. (1964), 'Some problems in particle technology: A statistical mechanical formulation', *Chemical Engineering Science* **19**(8), 555–574.
- Iddir, H. and Arastoopour, H. (2005), 'Modeling of multitype particle flow using the kinetic theory approach', *AIChE Journal* **51**(6), 1620–1632.
- Jackson, R. (2000), *The dynamics of fluidized particles*, Cambridge University Press.
- Kapur, P. (1972), 'Kinetics of granulation by non-random coalescence mechanism', *Chemical Engineering Science* **27**(10), 1863–1869.
- Kuipers, J., Van Duin, K., Van Beckum, F. and Van Swaaij, W. (1993), 'Computer simulation of the hydrodynamics of a two-dimensional gas-fluidized bed', *Computers & chemical engineering* **17**(8), 839–858.
- Kunii, D. and Levenspiel, O. (2013), *Fluidization engineering*, Elsevier.
- Lanczos, C. (1988), *Applied analysis*, Courier Corporation.
- Lebowitz, J. (1964), 'Exact solution of generalized percus-yevick equation for a mixture of hard spheres', *Physical Review* **133**(4A), A895.
- Lee, K. and Matsoukas, T. (2000), 'Simultaneous coagulation and break-up using constant-n monte carlo', *Powder Technology* **110**(1), 82–89.
- Lettieri, P. (2014), 'Fluid particles systems: Course notes', *Department of Chemical Engineering, University College London*.
- Lettieri, P., Cammarata, L., Micale, G. D. and Yates, J. (2002), 'Cfd simulations of gas fluidized beds using alternative eulerian-eulerian modelling approaches', *International Journal of Chemical Reactor Engineering* **1**(1).

- Lettieri, P., Saccone, G. and Cammarata, L. (2004), 'Predicting the transition from bubbling to slugging fluidization using computational fluid dynamics', *Chemical Engineering Research and Design* **82**(8), 939–944.
- Levenspiel, O. (1999), 'Chemical reaction engineering', *Industrial & engineering chemistry research* **38**(11), 4140–4143.
- Li, J., Luo, Z.-H., Lan, X.-Y., Xu, C.-M. and Gao, J.-S. (2013), 'Numerical simulation of the turbulent gas–solid flow and reaction in a polydisperse fcc riser reactor', *Powder technology* **237**, 569–580.
- Lipschutz, S. and Lipson, M. L. (2001), *Schaum's outline of theory and problems of linear algebra*, McGraw-Hill.
- Lun, C., Savage, S. B., Jeffrey, D. and Chepurniy, N. (1984), 'Kinetic theories for granular flow: inelastic particles in couette flow and slightly inelastic particles in a general flowfield', *Journal of fluid mechanics* **140**, 223–256.
- Marchisio, D. and Fox, R. (2003), 'Direct quadrature method of moments: Derivation, analysis and applications', *Journal of Computational Physics* .
- Marchisio, D. L. and Fox, R. O. (2005), 'Solution of population balance equations using the direct quadrature method of moments', *Journal of Aerosol Science* **36**(1), 43–73.
- Marchisio, D. L. and Fox, R. O. (2013), *Computational models for polydisperse particulate and multiphase systems*, Cambridge University Press.
- Marchisio, D. L., Pikturna, J. T., Fox, R. O., Vigil, R. D. and Barresi, A. A. (2003c), 'Quadrature method of moments for population-balance equations', *AIChE Journal* **49**(5), 1266–1276.
- Marchisio, D. L., Vigil, R. D. and Fox, R. O. (2003a), 'Implementation of the quadrature method of moments in cfd codes for aggregation–breakage problems', *Chemical Engineering Science* **58**(15), 3337–3351.
- Marchisio, D. L., Vigil, R. D. and Fox, R. O. (2003b), 'Quadrature method of moments for aggregation–breakage processes', *Journal of Colloid and Interface Science* **258**(2), 322–334.
- Marzocchella, A., Salatino, P., Di Pastena, V. and Lirer, L. (2000), 'Transient fluidization and segregation of binary mixtures of particles', *AIChE Journal* **46**(11), 2175–2182.
- Massimilla, L., Donsi, G. and Zucchini, C. (1972), 'The structure of bubble-free gas fluidized beds of fine fluid cracking catalyst particles', *Chemical Engineering Science* **27**(11), 2005–2015.
- Mazzei, L. (2008), Eulerian modelling and computational fluid dynamics simulation of mono and polydisperse fluidized suspension, PhD thesis, UCL (University College London).

- Mazzei, L. (2011), 'Limitations of quadrature-based moment methods for modeling inhomogeneous polydisperse fluidized powders', *Chemical engineering science* **66**(16), 3628–3640.
- Mazzei, L., Casillo, A., Lettieri, P. and Salatino, P. (2010), 'Cfd simulations of segregating fluidized bidisperse mixtures of particles differing in size', *Chemical Engineering Journal* **156**(2), 432–445.
- Mazzei, L. and Lettieri, P. (2008), 'Cfd simulations of expanding/contracting homogeneous fluidized beds and their transition to bubbling', *Chemical Engineering Science* **63**(24), 5831–5847.
- Mazzei, L., Marchisio, D. L. and Lettieri, P. (2009), 'Direct quadrature method of moments for the mixing of inert polydisperse fluidized powders and the role of numerical diffusion', *Industrial & Engineering Chemistry Research* **49**(11), 5141–5152.
- Mazzei, L., Marchisio, D. L. and Lettieri, P. (2012), 'New quadrature-based moment method for the mixing of inert polydisperse fluidized powders in commercial cfd codes', *AIChE Journal* **58**(10), 3054–3069.
- McGraw, R. (1997), 'Description of aerosol dynamics by the quadrature method of moments', *Aerosol Science and Technology* **27**(2), 255–265.
- McGraw, R. (2006), 'Correcting moment sequences for errors associated with advective transport'.
- McGraw, R. (2012), Correcting transport errors during advection of aerosol and cloud moment sequences in eulerian models, in 'Climate Models', InTech.
- Mutsers, S. and Rietema, K. (1977), 'The effect of interparticle forces on the expansion of a homogeneous gas-fluidized bed', *Powder Technology* **18**(2), 239–248.
- Narni, N. R., Warnecke, G., Kumar, J., Peglow, M. and Heinrich, S. (2012), Some modelling aspects of aggregation kernels and the aggregation population balance equations, in 'Eco-friendly Computing and Communication Systems', Springer, pp. 319–326.
- Nienow, A., Naimier, N. and Chiba, T. (1987), 'Studies of segregation/mixing in fluidised beds of different size particles', *Chemical Engineering Communications* **62**(1-6), 53–66.
- Olivieri, G., Marzocchella, A. and Salatino, P. (2004), 'Segregation of fluidized binary mixtures of granular solids', *AIChE journal* **50**(12), 3095–3106.
- Owoyemi, O., Mazzei, L. and Lettieri, P. (2007), 'Cfd modeling of binary-fluidized suspensions and investigation of role of particle–particle drag on mixing and segregation', *AIChE journal* **53**(8), 1924–1940.
- Pain, C., Mansoorzadeh, S. and De Oliveira, C. (2001), 'A study of bubbling and slugging fluidised beds using the two-fluid granular temperature model', *International Journal of Multiphase Flow* **27**(3), 527–551.

- Peglow, M. (2005), Beitrag zur Modellbildung von eigenschaftsverteiltern dispersen Systemen am Beispiel der Wirbelschicht-Sprühagglomeration, PhD thesis, Otto-von-Guericke-Universität Magdeburg, Universitätsbibliothek.
- Petitti, M., Nasuti, A., Marchisio, D. L., Vanni, M., Baldi, G., Mancini, N. and Podenzani, F. (2010), 'Bubble size distribution modeling in stirred gas-liquid reactors with qmom augmented by a new correction algorithm', *AIChE journal* **56**(1), 36–53.
- Petrovsky, I. (1957), 'Integral equations', *Eng. trans. H. Kamel and H. Komm*(Graylock Press, Rochester, NY, 1957) .
- Press, W., Teukolsky, S., Vetterling, W. and Flannery, B. (2002), 'Numerical recipes in c++: the art of scientific computing by william h', *Press. xxviii* .
- Qiaoqun, S., Huilin, L., Wentie, L., Yurong, H., Lidan, Y. and Gidaspow, D. (2005), 'Simulation and experiment of segregating/mixing of rice husk-sand mixture in a bubbling fluidized bed', *Fuel* **84**(14), 1739–1748.
- Ramkrishna, D. (2000), *Population balances: Theory and applications to particulate systems in engineering*, Academic press.
- Randolph, A. and Larson, M. (1971), 'Theory of particulate processes'.
- Rietema, K. and Piepers, H. (1990), 'The effect of interparticle forces on the stability of gas-fluidized beds—i. experimental evidence', *Chemical Engineering Science* **45**(6), 1627–1639.
- Rosner, D. E. and Yu, S. (2001), 'Mc simulation of aerosol aggregation and simultaneous spheroidization', *AIChE journal* **47**(3), 545–561.
- Sastry, K. V. (1975), 'Similarity size distribution of agglomerates during their growth by coalescence in granulation or green pelletization', *International Journal of Mineral Processing* **2**(2), 187–203.
- Schaafsma, S., Marx, T. and Hoffmann, A. (2006), 'Investigation of the particle flow-pattern and segregation in tapered fluidized bed granulators', *Chemical Engineering Science* **61**(14), 4467–4475.
- Schaeffer, D. G. (1987), 'Instability in the evolution equations describing incompressible granular flow', *Journal of differential equations* **66**(1), 19–50.
- Sinclair, J. and Jackson, R. (1989), 'Gas-particle flow in a vertical pipe with particle-particle interactions', *AIChE Journal* **35**(9), 1473–1486.
- Smith, M. and Matsoukas, T. (1998), 'Constant-number monte carlo simulation of population balances', *Chemical Engineering Science* **53**(9), 1777–1786.
- Smoluchowski, M. (1918), 'Towards a mathematical theory of the coagulation kinetics of colloidal solutions', *Z. Phys. Chem* **92**, 129–168.

- Syamlal, M., Rogers, W. and O'Brien, T. J. (1993), 'Mfix documentation: Theory guide', *National Energy Technology Laboratory, Department of Energy, Technical Note DOE/METC-95/1013 and NTIS/DE95000031*.
- Tan, H., Goldschmidt, M., Boerefijn, R., Hounslow, M., Salman, A. and Kuipers, J. (2004), 'Building population balance model for fluidized bed melt granulation: lessons from kinetic theory of granular flow', *Powder Technology* **142**(2), 103–109.
- Van Wachem, B., Schouten, J., Van den Bleek, C., Krishna, R. and Sinclair, J. (2001), 'Cfd modeling of gas-fluidized beds with a bimodal particle mixture', *AIChE Journal* **47**(6), 1292–1302.
- Vikas, V., Wang, Z., Fox, R. O. and Passalacqua, A. (2010), Development of high-order realizable finite-volume schemes for quadrature-based moment method, in '48th AIAA Aerospace Sciences Meeting Including the New Horizons Forum and Aerospace Exposition', p. 1080.
- Wirsum, M., Fett, F., Iwanowa, N. and Lukjanow, G. (2001), 'Particle mixing in bubbling fluidized beds of binary particle systems', *Powder Technology* **120**(1), 63–69.
- Wormsbecker, M., van Ommen, R., Nijenhuis, J., Tanfara, H. and Pugsley, T. (2009), 'The influence of vessel geometry on fluidized bed dryer hydrodynamics', *Powder Technology* **194**(1), 115–125.
- Wright, D. L. (2007), 'Numerical advection of moments of the particle size distribution in eulerian models', *Journal of Aerosol Science* **38**(3), 352–369.
- Yan, W.-C., Luo, Z.-H., Lu, Y.-H. and Chen, X.-D. (2012), 'A cfd-pbm-pmlm integrated model for the gas–solid flow fields in fluidized bed polymerization reactors', *AIChE journal* **58**(6), 1717–1732.
- Yao, Y., Su, J.-W. and Luo, Z.-H. (2015), 'Cfd-pbm modeling polydisperse polymerization fbrs with simultaneous particle growth and aggregation: the effect of the method of moments', *Powder Technology* **272**, 142–152.
- Zhang, Y., Zhong, W., Jin, B. and Xiao, R. (2012), 'Mixing and segregation behavior in a spout-fluid bed: effect of particle size', *Industrial & Engineering Chemistry Research* **51**(43), 14247–14257.