MULTI-COMPONENT POPULATION BALANCE MODELING OF WET GRANULATION VIA CONSTANT-NUMBER MONTE CARLO

A Dissertation in
Chemical Engineering
by
Carl L. Marshall, Jr.

© 2012 Carl L. Marshall, Jr.

Submitted in Partial Fulfillment
of the Requirements
for the Degree of

Doctor of Philosophy

May 2012
The dissertation of Carl L. Marshall, Jr. was reviewed and approved* by the following:

Themis Matsoukas  
Professor of Chemical Engineering  
Dissertation Advisor, Chair of Committee

Ali Borhan  
Professor of Chemical Engineering

Virendra Puri  
Distinguished Professor of Agricultural and Biological Engineering

Darrell Velegol  
Professor of Chemical Engineering

Andrew Zydney  
Walter L. Robb Chair, Professor, and Department Head of Chemical Engineering

*Signatures are on file in the Graduate School.
Abstract

Fluid bed granulation is a complex, multi-phase, multi-component unit operation often used in the pharmaceutical industry. Various attributes such as particle morphology, size, porosity, wettability, and binder viscosity play key roles in determining the overall size and composition of the granules that are produced. Many researchers have sought to investigate, from both an experimental and theoretical perspective, the key parameters that impact granulation. The research presented within this dissertation follows suit.

In this work, the concept of a multi-component population balance equation was utilized as a mathematical framework for modeling fluidized bed granulation. Ultimately, it was desired to develop a computational model that can simultaneously consider the evolution of granule size, moisture content, and compositional distribution. Specifically, the end goal was to develop a model that can evaluate the dynamic evolution of a four-component system containing two powders, binder polymer content, and binder moisture content. To accomplish this task, a very systematic approach was taken.

First, three different numerical methodologies for the population balance equation were examined: a “rigorous” discrete method, the direct quadrature method of moments (DQMOM) technique, and constant-number Monte Carlo. The system under examination contained two morphologically distinct powders that were assumed to already have binder present on the surface. The rigorous discrete method was found to be very computationally efficient. DQMOM was found to be very fast, but was unreliable when analyzing kernels with significant compositional dependence. Constant-number Monte Carlo was determined to be the best choice when analyzing multi-dimensional, composition-dependent population balance equations.

Any model that is developed, though complex, is of no practical good if it cannot faithfully recreate experimental results. As such, simulation results were
compared with experimental results at nearly every step along the way in this research. It was found that the constant-number Monte Carlo model continually matches the experimental data for three granulation cases of increasing difficulty: a two-component case that considers the continuous addition of binder to a granulator, a three-component case that evaluates the impact of evaporation and moisture content on the granule growth profile, and ultimately the four-component case mentioned above where granule growth, moisture content, binder distribution, and composition distribution are evaluated.

An effective pharmaceutical wet granulation model should include at least three components: (i) a population balance methodology that tracks the distribution and composition of all species of interest, (ii) a physically based description of agglomeration and breakage, and (iii) hydrodynamic modeling. While the third piece of the puzzle was beyond the current scope of work, significant strides were made in regards to the other two. It was shown that the constant-number Monte Carlo methodology can utilize a complex morphologically-based compositional-dependent coagulation kernel (criteria ii) and track the size evolution of two different powders while also considering the impact of moisture loss on the system (criteria i). Thus, the development of the constant-number Monte Carlo model contained within this dissertation helps to fill a critical need within the granulation community for a model that can provide a thorough analysis of multi-component granulation problems.
Table of Contents

List of Figures ix
List of Tables xii
List of Symbols xiii
Acknowledgments xvii

Chapter 1 Introduction 1
1.1 Granulation Overview ............................ 1
1.2 Pharmaceutical Granulation ...................... 4
1.3 Granulation and the F.D.A. ...................... 6
1.4 Research Objectives ............................ 7
1.5 Thesis Outline ................................ 8

Chapter 2 Background 10
2.1 Chapter Overview ................................ 10
2.2 Population Balance Description .................. 10
2.3 Multi-dimensional/Multi-component Population Balance Equations 12
2.4 Kernel Overview ................................ 14
   2.4.1 KTGF Kernel ................................. 16
   2.4.2 Kernel Success Factor ....................... 19
2.5 Population Balance Solution Methods ............. 23
   2.5.1 Sectionalization Techniques ................ 23
   2.5.2 Method of Moments .......................... 24
   2.5.3 Stochastic Techniques ....................... 26
   2.5.4 Development of Constant-Number Monte Carlo .... 27
      2.5.4.1 Concentrations ......................... 28
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.5.4.2</td>
<td>Time Increment</td>
<td>29</td>
</tr>
<tr>
<td>2.5.4.3</td>
<td>Characteristic Dimensional Variables</td>
<td>29</td>
</tr>
<tr>
<td>2.6</td>
<td>Derivation of Equations</td>
<td>30</td>
</tr>
<tr>
<td>2.6.1</td>
<td>Dimensionless KTGF Kernel Derivation</td>
<td>30</td>
</tr>
<tr>
<td>2.6.2</td>
<td>Dimensionless Stokes Number Derivation</td>
<td>32</td>
</tr>
<tr>
<td>2.6.3</td>
<td>Dimensionless Coagulation Rate Derivation</td>
<td>33</td>
</tr>
<tr>
<td>2.6.4</td>
<td>Dimensionless Time Increment Derivation</td>
<td>35</td>
</tr>
</tbody>
</table>

Chapter 3 Numerical Simulations of Two-Component Granulation: Comparison of Three Methods

<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1</td>
<td>Chapter Overview</td>
<td>37</td>
</tr>
<tr>
<td>3.2</td>
<td>Multi-component Granulation and Composition</td>
<td>38</td>
</tr>
<tr>
<td>3.3</td>
<td>Theoretical Background</td>
<td>39</td>
</tr>
<tr>
<td>3.4</td>
<td>Moments</td>
<td>40</td>
</tr>
<tr>
<td>3.5</td>
<td>Kernels</td>
<td>41</td>
</tr>
<tr>
<td>3.6</td>
<td>Methods of Solution</td>
<td>43</td>
</tr>
<tr>
<td>3.6.1</td>
<td>Rigorous Discrete Method</td>
<td>43</td>
</tr>
<tr>
<td>3.6.2</td>
<td>Constant-Number Monte Carlo (cNMC)</td>
<td>44</td>
</tr>
<tr>
<td>3.6.3</td>
<td>DQMOM</td>
<td>44</td>
</tr>
<tr>
<td>3.7</td>
<td>Results</td>
<td>46</td>
</tr>
<tr>
<td>3.8</td>
<td>Discussion</td>
<td>49</td>
</tr>
<tr>
<td>3.9</td>
<td>Conclusions</td>
<td>54</td>
</tr>
</tbody>
</table>

Chapter 4 Multicomponent Population Balance Modeling of Granulation with Continuous Addition of Binder

<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.1</td>
<td>Chapter Overview</td>
<td>56</td>
</tr>
<tr>
<td>4.2</td>
<td>Multi-Component Population Balance Model Development</td>
<td>57</td>
</tr>
<tr>
<td>4.2.1</td>
<td>Kernel Description</td>
<td>58</td>
</tr>
<tr>
<td>4.2.2</td>
<td>Composition-Dependent Portion of Kernel</td>
<td>59</td>
</tr>
<tr>
<td>4.3</td>
<td>Development of Constant-Number Monte Carlo with a Flow Stream</td>
<td>59</td>
</tr>
<tr>
<td>4.3.1</td>
<td>Event Rates and Probabilities</td>
<td>60</td>
</tr>
<tr>
<td>4.3.2</td>
<td>Event Implementation</td>
<td>61</td>
</tr>
<tr>
<td>4.3.3</td>
<td>Concentrations</td>
<td>62</td>
</tr>
<tr>
<td>4.3.4</td>
<td>Time Increment</td>
<td>62</td>
</tr>
<tr>
<td>4.4</td>
<td>Validation of cNMC algorithm</td>
<td>62</td>
</tr>
<tr>
<td>4.4.1</td>
<td>Case 1: Constant Kernel</td>
<td>63</td>
</tr>
<tr>
<td>4.4.2</td>
<td>Case 2: KTGF Kernel</td>
<td>65</td>
</tr>
<tr>
<td>4.5</td>
<td>Results and Discussion</td>
<td>66</td>
</tr>
<tr>
<td>4.5.1</td>
<td>$K_0$ Analysis</td>
<td>68</td>
</tr>
<tr>
<td>4.5.2</td>
<td>Binder Distribution Analysis</td>
<td>69</td>
</tr>
</tbody>
</table>
4.5.3  Coarsening Factor Analysis ........................................ 71
4.6  Conclusions ............................................................. 72

Chapter 5  Continuous Addition of Binder with Moisture Removal 74
5.1  Chapter Overview ......................................................... 74
5.2  Theoretical Background ................................................ 75
  5.2.1  Governing Equations .............................................. 76
  5.2.2  Incorporation of Binder Drying Effect .......................... 79
  5.2.3  Evaporation Rate .................................................... 80
5.3  Development of Constant-Number Monte Carlo with Evaporation . 83
  5.3.1  Event Rates and Probabilities .................................... 83
    5.3.1.1  Coagulation Event ........................................... 84
    5.3.1.2  Evaporation Event ........................................... 84
    5.3.1.3  Flow In Event ................................................ 85
    5.3.1.4  Flow Out Event ............................................... 86
    5.3.1.5  Event Probability ............................................. 86
  5.3.2  Event Implementation ............................................. 86
  5.3.3  Concentrations ..................................................... 88
  5.3.4  Time Increment .................................................... 89
5.4  Derivation of Equations ................................................ 89
  5.4.1  Dimensionless Change in Number Concentration Derivation . 89
  5.4.2  Dimensionless Change in Particle Mass Concentration Deriva-
        tion ................................................................. 90
  5.4.3  Dimensionless Change in Moisture Mass Concentration Deriva-
        tion ................................................................. 91
  5.4.4  Dimensionless Evaporation Rate Derivation .................... 92
  5.4.5  Dimensionless Monte Carlo Evaporation Event Rate Deriva-
        tion ................................................................. 93
  5.4.6  Dimensionless Monte Carlo Flow In Event Rate Derivation ... 93
5.5  Validation of cNMC algorithm ......................................... 94
  5.5.1  Comparison of Monte Carlo with Mathematica .................. 95
  5.5.2  Comparison of Monte Carlo with the Rigorous Discrete Method 97
5.6  Results and Discussion ............................................... 99
  5.6.1  Temperature and Granulation ..................................... 99
  5.6.2  Binder Distribution ............................................... 101
    5.6.2.1  Effect of Temperature ..................................... 102
    5.6.2.2  Effect of Monte Carlo Evaporation Mass .................. 103
  5.6.3  Comparison with Experiment ..................................... 106
5.7  Conclusions ............................................................. 109
List of Figures

1.1 Schematic of the granulation of two particles. Powder 1 is pink, powder 2 is purple, and the liquid binder is represented by green spheres. The relative size of the particles and droplets is not drawn to scale. ................................. 1
1.2 Schematic of three different types of granulators. ................. 3
1.3 Plug flow diagram of the formulation process for the production of tablets or capsules. Granulation is the first step in the process. . . . 5

2.1 Application of collision theory for derivation of size-dependent portion of KTGF kernel .................................................. 17
2.2 Possible outcomes of a collision. Particle A is red, particle B is purple, and the binder coating is green. Figure is adapted from [1]. 20
2.3 Optical pictures of the three excipients studied in this dissertation. The scale bar in the images is 1,000 µm. Optical images obtained from Stepanek et al. [2]. ........................................... 21

3.1 Constant kernel with no compositional dependence. Initial seeds bidisperse, \( v_A = 1, v_B = 10 \). .................................................. 47
3.2 KTGF kernel – no compositional dependence (with \( \gamma = 0 \)). Initial seeds bidisperse, \( v_A = 1, v_B = 10 \). .............................. 48
3.3 Constant kernel with compositional dependence. Initial seeds bidisperse, \( v_A = 1, v_B = 10 \). .................................................. 50
3.4 KTGF kernel (\( \gamma = 0 \)) with compositional dependence. Initial seeds bidisperse, \( v_A = 1, v_B = 10 \). .................................................. 51
3.5 KTGF kernel (\( \gamma = 1 \)) with compositional dependence. Initial condition bidisperse with \( v_A = 1, v_B = 10 \). .............................. 52
3.6 KTGF kernel (\( \gamma = 1 \)) with compositional dependence. Initial condition bidisperse with \( v_A = 2, v_B = 20 \). .............................. 53
3.7 KTGF kernel (\( \gamma = 1 \)) with compositional dependence. Initial condition: solute particles are monodisperse with \( v_A = 2 \); excipient particles follow normal distribution with \( \bar{v}_B = 20 \) and variance \( \sigma^2 = 25 \). 54
3.8 KTGF kernel (γ = 1) with compositional dependence. Initial condition: solute particles are monodisperse with \( v_A = 2 \); excipient particles follow lognormal distribution with \( \bar{v}_B = 23 \), and geometric standard deviation \( s_g = 1.65 \). ............................................. 55

4.1 Average granule size versus time for constant kernel. Average size was made dimensionless by monodisperse binder droplet size. Binder-binder collisions were permitted. ........................................ 64

4.2 Average granule size versus time for KTGF kernel with geometric success factor. Average size was made dimensionless by monodisperse binder droplet size. Binder-binder collisions were prohibited. Particle-binder \( K_0 \) is 100 times greater than particle-particle \( K_0 \). ... 65

4.3 Comparison of Experimental \( d_{43} \) with Simulation \( d_{43} \) ............... 68

4.4 Binder distribution as a function of time. The binder mass fraction axis is linear and ranges from 0 to 1, whereas the size axis is logarithmic and ranges from a dimensionless size of 0 to approximately 20,000 (\( d_{43} \approx 1,630 \) µm). For each size class, the height of each bar is only relative to the other bars within that size class. ............... 70

4.5 Analysis of coarsening factor (number of binder droplets introduced per event) impact on simulation speed and accuracy for Mannitol. A typical simulation run time for several different coarsening factors is given in 4.5a. Average granulation growth profiles for several coarsening factors are shown in 4.5b (average of 10 simulation runs). 73

5.1 Schematic of the four processes considered during granulation with evaporation. ............................................. 76

5.2 Comparison of Monte Carlo particle mass concentration results with Mathematica results for a very high evaporation rate. ................. 95

5.3 Comparison of Monte Carlo moisture mass concentration results with Mathematica results for very high, moderate, and very low evaporation rates. ................................................. 96

5.4 Comparison of average binder surface coverage as calculated via Monte Carlo and rigorous discrete (RM) analytical equation. .... 97

5.5 Comparison of evaporation rate as calculated via Monte Carlo and rigorous discrete (RM) method. ................................. 98

5.6 Comparison of outlet air moisture content as calculated via Monte Carlo and rigorous discrete (RM) method. ................................. 98

5.7 Impact of temperature on granule growth profile and moisture content during Mannitol granulation. ........................................ 101
5.8 Impact of increased evaporation rate on granule growth profile and moisture content. ......................................................... 102
5.9 Analysis of mass fraction content for binder (HPC and water) relative to granule mass and HPC relative to binder mass (HPC and water). ........................................................................ 103
5.10 Analysis of mass fraction content for HPC relative to total binder mass (HPC and water) ....................................................... 104
5.11 Impact of Monte Carlo evaporation mass on granule growth profile and moisture content. ......................................................... 105
5.12 Comparison of Monte Carlo simulations with Mannitol experimental data for granule growth profile and moisture content. .... 108
5.13 Comparison of Monte Carlo simulations with Avicel experimental data for granule growth profile and moisture content. ........ 108

6.1 Analysis of experimental A-TAB mass fraction vs. sieve cut for granulation of a 50:50 wt% mixture of Mannitol and A-TAB. Figure reproduced from Rajniak et al. [3] with permission. ............... 111
6.2 Comparison of Monte Carlo simulations with experimental data for granule growth profile and moisture content of 50:50 wt% Mannitol and A-TAB. ................................................................. 117
6.3 Analysis of simulation A-TAB mass fraction vs. sieve cut for granulation of a 50:50 wt% mixture of Mannitol and A-TAB. The red line indicates the ideal granule composition of 50 weight % A-TAB across all sizes. ................................................................. 117
6.4 Analysis of A-TAB mass fraction and HPC mass fraction as a function of size after 15 minutes of granulating a 50:50 wt% mixture of Mannitol and A-TAB. ................................................................. 118
6.5 Analysis of A-TAB mass fraction and HPC mass fraction as a function of size after 30 minutes of granulating a 50:50 wt% mixture of Mannitol and A-TAB. ................................................................. 119
6.6 Analysis of A-TAB mass fraction and HPC mass fraction as a function of size after 45 minutes of granulating a 50:50 wt% mixture of Mannitol and A-TAB. ................................................................. 119
6.7 Analysis of A-TAB mass fraction and HPC mass fraction as a function of size after 55 minutes of granulating a 50:50 wt% mixture of Mannitol and A-TAB. ................................................................. 120
### List of Tables

2.1 Coalescence Kernels .................................................. 16
2.2 Particle shape descriptors for experimental excipients ............ 22
2.3 Key constant-number Monte Carlo equations for coagulation ... 30

3.1 Summary of cases ..................................................... 41

4.1 Key constant-number Monte Carlo equations ....................... 63
4.2 Simulation parameters .................................................. 67
4.3 Optimized values of $K_0$ ............................................. 69

5.1 Dimensionless variables ............................................... 78
5.2 Key constant-number Monte Carlo equations for granulation with evaporation ................................................. 87
5.3 Simulation parameters for Monte Carlo granulation with evaporation .................................................. 100
5.4 Impact of Monte Carlo Evaporation Mass $w$ on Simulation Times . 105
5.5 Optimized values of $K_0$ and $\omega$ .................................. 109

6.1 Simulation parameters for Monte Carlo granulation with evaporation 116
### List of Symbols

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Definition</th>
<th>Page/Section</th>
</tr>
</thead>
<tbody>
<tr>
<td>$B_c$</td>
<td>Mass fraction of binder HPC relative to binder HPC and water, dimensionless</td>
<td>80</td>
</tr>
<tr>
<td>$B_c^*$</td>
<td>Critical binder mass fraction value, dimensionless</td>
<td>80</td>
</tr>
<tr>
<td>$b$</td>
<td>Empirical correction parameter used in calculation of $\eta$, dimensionless</td>
<td>21</td>
</tr>
<tr>
<td>$C_0$</td>
<td>Initial particle number concentration, $1/m^3$</td>
<td>29</td>
</tr>
<tr>
<td>$C_p$</td>
<td>Particle number concentration, $1/m^3$</td>
<td>28</td>
</tr>
<tr>
<td>$C_*$</td>
<td>Characteristic concentration = $C_0$, $1/m^3$</td>
<td>28</td>
</tr>
<tr>
<td>$c$</td>
<td>Empirical correction parameter used in calculation of $\eta$, dimensionless</td>
<td>21</td>
</tr>
<tr>
<td>$c_m$</td>
<td>Dimensionless particle mass concentration</td>
<td>28</td>
</tr>
<tr>
<td>$c_p$</td>
<td>Dimensionless particle number concentration</td>
<td>28</td>
</tr>
<tr>
<td>$d_i$</td>
<td>Diameter of particle $i$, m</td>
<td>18</td>
</tr>
<tr>
<td>$d_b$</td>
<td>Characteristic diameter used to form dimensionless diameter, m</td>
<td>29</td>
</tr>
<tr>
<td>$d_{43}$</td>
<td>Population volume mean diameter, m</td>
<td>18</td>
</tr>
<tr>
<td>$F(V,U)dVdU$</td>
<td>Number concentration of particles with total mass $V$ and binder mass $U$, $1/m^3$</td>
<td>14</td>
</tr>
<tr>
<td>$f$</td>
<td>Empirical correction parameter used in calculation of $\phi$, dimensionless</td>
<td>23</td>
</tr>
</tbody>
</table>
g  Empirical correction parameter used in calculation of \( \phi \),
dimensionless, p. 23

\( h_a \)  Asperity (roughness) of the primary particle surface, m,
p. 22

\( K(V_1,U_1;V_2,U_2) = K_{12} \)  Coagulation kernel between particles 1 and 2, m\(^3\)/s, p. 14

\( K_0 \)  Coagulation constant, m\(^3\)/s, p. 19

\( k_{12} \)  Dimensionless part of coagulation kernel between parti-
cles 1 and 2, p. 19

\( \langle k_{12} \rangle \)  Dimensionless ensemble average of kernels within the
population, p. 28

\( N \)  Number of particles in simulation box, dimensionless,
p. 28

\( n \)  Number of binder droplets added per Monte Carlo event,
dimensionless, p. 61

\( P_{\text{event}} \)  Probability of a Monte Carlo event, dimensionless, p. 60

\( \dot{Q} \)  Binder droplet flow rate, 1/s, p. 61

\( r_{\text{event}} \)  Dimensionless rate of a Monte Carlo event, p. 60

\( r_g \)  Mean radius of gyration, m, p. 23

\( \text{St} \)  Stokes number used in calculation of physical portion of
physical success factor, dimensionless, p. 22

\( \text{St}^* \)  Critical Stokes number, dimensionless, p. 22

\( t \)  Time, s, p. 57

\( t_s \)  Characteristic time = \( \tau_{\text{coag}} \), s, p. 28

\( U_i \)  Mass of binder in particle \( i \), kg, p. 14

\( u_0 \)  Velocity of particle collision, m/s, p. 22

\( V_i \)  Total mass of particle \( i \), kg, p. 11
$V_b$ Characteristic volume of a monodisperse binder droplet with diameter $d_b$, m$^3$, p. 31

$V_R$ Reactor volume, m$^3$, p. 61

$V_*$ Characteristic volume $= 1/C_0$, m$^3$, p. 61

$\bar{v}$ Average dimensionless mass in simulation box, p. 28

$v_b$ Dimensionless mass of a monodisperse binder droplet with diameter $d_b$, p. 62

$v_i$ Total dimensionless volume of particle $i$, p. 18

$\delta v_{\text{event}}$ Change in dimensionless mass due to particulate event, p. 28

$v_m$ Total dimensionless mass of particle $i$, p. 106

$w$ Dimensionless evaporation mass, p. 84

$x_i$ Dimensionless diameter of particle $i$, p. 18

$x_{43}$ Dimensionless population volume mean diameter, p. 58

$y_i$ Binder to solid ratio of particle $i$, dimensionless, p. 21

$z_{12}$ Portion of dimensionless coagulation kernel between particles 1 and 2, p. 58

**Greek symbols**

$\beta$ Agglomeration constant, $1/(m^{\gamma-2.5} s)$, p. 18

$\gamma$ Exponent, dimensionless, p. 18

$\eta_i$ Accessible binder fraction of particle $i$, dimensionless, p. 20

$\lambda_{12}$ Binder layer thickness between 2 colliding particles, m, p. 22

$\mu$ Binder viscosity, kg/(m s), p. 22

$\rho$ Particle density, kg/m$^3$, p. 22
\( \tau \) Dimensionless time, p. 29

\( \tau_{\text{coag}} \) Characteristic coagulation time, s, p. 28

\( \tau_{\text{evap}} \) Characteristic evaporation time, s, p. 92

\( \tau_{\text{in}} \) Characteristic flow in time, s, p. 78

\( \tau_{\text{FS}} \) Characteristic binder flow stream time, s, p. 61

\( \tau_{\text{out}} \) Characteristic flow out time, s, p. 78

\( \phi_i \) Displaced binder volume of particle \( i \), dimensionless, p. 22

\( \psi_{12} \) Collision success factor, dimensionless, p. 18

\( \psi_{\text{fresh}} \) Geometric portion of collision success factor, dimensionless, p. 80

\( \psi_{\text{geom}} \) Geometric portion of collision success factor, dimensionless, p. 20

\( \psi_{\text{phys}} \) Physical portion of collision success factor, dimensionless, p. 20

\( \omega \) Evaporation rate coefficient, dimensionless, p. 81

**Sub- and superscripts**

coag Coagulation, p. 60

FS Binder flow stream, p. 60

old Pre-Monte Carlo event value, p. 28
Acknowledgments

I would like to thank my Penn State advisor, Themis Matsoukas, for all his help and guidance. I truly believe that I could not have had a better advisor. Your keen awareness of thermodynamics and population balances is amazing as is your often gentle spirit (unless of course it is locked in battle with some evil Microsoft file that I asked you to review). Thank you for helping me become a better writer and speaker. Thank you for your encouragement and for believing in me. Thank you for this opportunity!

I would also like to thank my industrial collaborator and co-advisor, Pavol Rajniak, for his help, encouragement and humor throughout this endeavor. You were always in good spirits and willing to lend a helping hand or a listening ear when I called asking about some specific granulation question. I look forward to our work together as colleagues.

Thank you to Janna Maranas and our department head, Andrew Zydney, for taking a chance on someone who had been away from academia for over 7 years and supporting me as a PhD candidate.

I would like to acknowledge my financial support: the NSF Grant Opportunities for Academic Liaison with Industry (GOALI) Grant No CBET-0651644.

Thank you to Walter R. and Aura Lee Supina who provided a graduate fellowship to me during my first semester at Penn State.

Thank you to my labmate, Anaram Shahravan, for the friendship and support during the last several years. You will not soon be forgotten. In fact, I hope we see each other often in the years ahead. Good luck as you finish your studies and move out to finally be with your husband! I don’t know how you did it for over four years.
Thank you to my thesis committee members: Ali Borhan, Virendra Puri, Darrell Velegol and (formerly) Patrick Cirino. I appreciate you taking the time to serve on my committee and provide me with suggestive feedback and sharpening my skills as an engineer.

Thank you to the current and former members of the departmental staff: Sue Ellen Bainbridge, Laurinda Benner, Stephen Black, Sandra Brown, Dave Decapria, Bre-anne Greenaway, Randy Ripka, Mary Jane Smith, Ruth Smith, and Steven Smith. You all have made my graduate life easier in one way or another and, for that, I am truly grateful.

Thank you to Pastor Randy Smyre for his encouragement and friendship during the years. Though we may move apart geographically, you will always remain close in my thoughts and prayers. I am truly honored to be counted as one of your friends.

Thank you to my children, Reissa and Kai. I know you sacrificed a lot of “daddy time” during the past 4 and a half years. I could not have asked for two better children. (Of course, that depends on what day you ask). Please know that I love you both very much. The most! Ha, I said it and now you can never take it back. It will live on in print forever.

To my wife and best friend, Noelle, without whom this task would not have been even remotely possible. Thank you for the encouragement, the prayers, and putting up with my semi-constant stressed out state especially as my Ph.D. days draw to a close. I could not have done this without you! A Grecian cruise is far too insufficient a way to even begin to say thanks...

Finally, I would like to thank my Lord and Savior Jesus Christ for the gifts and talents He has given me and for the great group of people He has brought into my life.

“I have fought the good fight, I have finished the race, I have kept the faith.”
-2 Timothy 4:7
Chapter 1

Introduction

1.1 Granulation Overview

Granulation is a unit operation that combines smaller primary particles/granules to form larger granules in which the individual particles can still be identified (see Figure 1.1). Though this topic is not commonly studied in traditional undergraduate chemical engineering curriculums, the subject is encountered frequently enough to warrant a chapter in Perry’s Chemical Engineers’ Handbook [4]. Granulation is utilized across several industries including agricultural, food, mineral processing, pharmaceuticals, and specialty chemicals [5–8]. At the turn of the century, the annual value of powder/particulate production in the United States was estimated at $1 trillion [5]. The significant financial implications, in addition to the challenging scientific questions still unresolved in this field of study, demand an increased scrutiny on the fundamentals of such a common unit operation as granulation.

Several forms of granulation may be encountered in industry ranging from dry

Figure 1.1: Schematic of the granulation of two particles. Powder 1 is pink, powder 2 is purple, and the liquid binder is represented by green spheres. The relative size of the particles and droplets is not drawn to scale.
granulation (where a liquid binder, or glue-like substance, is not used) to various wet granulation techniques (where a liquid binder is used) such as drum granulation, high shear granulation, and fluidized bed granulation (see Figure 1.2). Regardless of the type of granulation conducted, the unit operation itself has many benefits including the minimization of dust formation, increased flowability, reduced segregation, and the promotion of uniform mixing. However, for many years the design and control of granulation processes were “considered more of an art than a science” [9] and were guided more by empiricism rather than an innate understanding of the underlying principles governing this unit operation [10]. Part of the reason for such empirical guidance was due to the complex interactions between the operating conditions and the material properties of the powders being granulated. Several variables can impact granulation. In a drum granulator, the angle of the bed and the rotary speed are critical design constraints [11]. In a fluidized bed granulator, critical variables include, but are not limited to, binder droplet size, binder nozzle type and location, binder viscosity, particle morphology, gas velocity, and granulation temperature [6]. Important variables during high shear granulation include the binder spray rate, the fill level and the agitator speed [6]. Due to the many complex interactions, “rules of thumb” and industry-specific best practices were often used to determine the best granulation approach. Fortunately, within the past few decades, there has been significant improvement in the understanding of granulation processes.

It is now generally understood that wet granulation is governed by several key processes [4,5]:

1. wetting and nucleation,
2. coalescence, mixing, and consolidation,
3. breakage and attrition.

Nucleation starts after binder is introduced and begins to wet the primary particles. This step may be characterized by a coating period as sufficient binder is deposited on the surface to promote granulation. Once nuclei are formed, coalescence and consolidation occur. Throughout this dissertation, the terms coalescence, aggregation, and granulation are used interchangeably to describe the combination of
(a) Drum granulator

(b) High Shear Granulator (from Perry’s Chemical Engineers’ Handbook [4])

(c) Wurster-style Fluidized Bed Granulator

Figure 1.2: Schematic of three different types of granulators.
smaller primary particles and/or granules to form granules of a larger size. Consolidation transpires when granules compact due to collisions with other granules and equipment surfaces (e.g., vessel walls or agitators); during consolidation, liquid binder present in the interstitial space of a granule may be squeezed to the surface. Larger granules are produced via the formation of liquid bridges between two colliding particles. Depending on the strength of the bridges, granule size reduction is also possible, either via breakage of wet granules or by fracture of dried granules in which the binder has solidified. As overall particle growth is the desired effect, the resulting population of granules is typically described by size. However, if the granulation process involves more than one component, granule composition is an additional dimension that characterizes the population. Such a multi-component analysis should be considered when studying wet granulation which has at least two components, solid powder and binder. This analysis is even more important when considering pharmaceutical wet granulation which generally includes at least three components: excipient, active pharmaceutical ingredient (API), and binder.

1.2 Pharmaceutical Granulation

The vast majority of pharmaceutical products destined for consumption in the United States is in the form of a tablet or capsule [12]. The formulation of a pharmaceutical tablet or capsule is schematically shown in Figure 1.3. After an API powder is produced, the first step in the formulation process is granulation. Besides the general benefits of granulation listed above, pharmaceutical-specific reasons also include [13]:

1. increasing the uniformity of drug distribution in the product,

2. densifying the material,

3. facilitating the metering or volumetric dispensing of the product, and

4. improving the appearance of the product.

The caveat to remember, however, is that poor granulation can result in inhomogeneous granules and such processing problems as downstream handling issues and poor tableting [5].
In the pharmaceutical industry, wet granulation is the more common operation relative to dry granulation. Common techniques include high shear mixing and fluidized bed granulation. High shear granulators are similar in structure to a standard kitchen blender with an impeller for mixing and a chopper for breaking clumps of material. As the impeller causes intimate mixing of the granules, consolidation of the granules occurs. Granules produced using this type of granulator are typically very dense and useful for applications where a time-delayed, controlled release of the drug is desired. Fluidized bed granulation, on the other hand, uses fluidizing air to suspend the bed and simultaneously dry the granules during granulation. The resulting granules tend to be more porous relative to high shear granulator granules and are beneficial in cases where a quick release of the drug is desired.

The research in this dissertation centers around granules produced using a Wurster-style granulator (see Figure 1.2c) [3,15]. Dry powder(s) is initially charged to the granulator and is then fluidized via heated air. This air enters the granulator through a perforated stainless steel plate with two different mesh sizes to permit faster flow in the center of the granulator and lower flow in the periphery. The screen also prevents the powder from exiting via the bottom whereas filters capture
any entrained solids attempting to exit the granulator with the air. Binder is fed to the system via atomizer nozzles. The Wurster unit differs from a conventional top-spray fluid bed granulator in that the atomizer nozzle for the binder is at the bottom of the unit and is oriented upwards. In addition, the Wurster unit contains a central cylinder that, in addition to permitting separate flow regimes, facilitates a regular circulation pattern and uniform distribution of binder. It thus has the potential to produce granules with a more uniform drug distribution and narrower particle size distribution [3]. In the pharmaceutical industry, this control is critical.

1.3 Granulation and the F.D.A.

Ideally, at the end of a multi-component granulation, each granule in the final product has the exact same composition. Practically, this ideal may be relaxed to permit a final product whose average composition across all size classes (sieve cuts) is uniform with an arbitrarily small standard deviation. However, recent research indicates that a component may appear at above-average concentrations in some size classes and below average in others [3,16–18]. This uneven distribution of components among granules suggests that granulation can be composition-dependent; it may also be inferred that there exists an inhomogeneous distribution of components not only among granules, but also within them [3]. Though this problem sounds like sufficient cause for alarm, its root causes have not been thoroughly investigated. The reason is that, despite the potential for granule inhomogeneity, granule formation is only an intermediate step in the formulation process; the end product (pharmaceutical tablets) often still meets the final content uniformity specifications [19]. However, the regulatory culture is now shifting to one that demands that not only final products, but also intermediate products such as granules, require a certain quality.

Partially due to concerns such as this, and partially due to the eternal chase for “better science”, the US Food and Drug Administration (FDA) is implementing a new paradigm entitled ”Quality by Design (QbD)” [20]. Under this initiative, the pharmaceutical industry is moving from an industry where quality is predominantly based on achieving a predetermined set of agreed-upon parameters to one that is based on functional, predictive models [21]. In 2000, The FDA published a
guidance entitled “Q6A Specifications: Test Procedures and Acceptance Criteria for New Drug Substances and New Drug Products: Chemical Substances.” In this guidance, “The quality of drug substances and drug products is determined by their design, development, in-process controls, GMP controls, process validation, and by specifications applied to them throughout development and manufacture” [22]. The general premise of these initiatives is a desire for the industry to not only pass established quality testing points, but to also make a more concerted effort to build quality into the process. As such, there is also an increased regulatory incentive to expand our understanding of the fundamentals of common unit operations such as granulation.

1.4 Research Objectives

The overarching goal of this dissertation is to develop a mathematical model that captures the key parameters affecting granulation. The model is developed using a mathematical tool known as a population balance equation (PBE). The specific goals that will be studied include:

1. Examining various numerical solution techniques for multi-component PBEs for speed and robustness.

2. Examining the impact of primary particle surface roughness on the granule growth rate and the subsequent binder distribution among granules of different sizes. A two-component PBE is required for this study.

3. Examining the impact of moisture removal on granule growth. A three-component PBE is required for this study.

4. Examining how the concurrent granulation of two solids with different particle morphologies affect the granule growth rate and compositional distribution across various size classes. A four-component PBE is required for this study.

The model is generally solved using a numerical technique known as constant-number Monte Carlo. For items 2-4 above, the simulation results are validated by comparison with experimental results where available.
1.5 Thesis Outline

This thesis is composed of seven chapters. In chapter 2, a general background is given that provides an overview of single and multi-component population balance equations (PBEs). A key component of granulation-focused PBEs centers on the choice of an aggregation kernel. As such, a discussion on the proper selection of a kernel is included. More detail on the rationale for selecting the KTGF (kinetic theory of granular flow) kernel as an appropriate kernel for modeling granulation in a fluidized bed granulator is given. The chapter concludes with a discussion on various numerical solution techniques for solving PBEs. A significant portion of this section is devoted to the technique featured throughout this dissertation, the constant-number Monte Carlo (cNMC) algorithm.

Chapter 3 is devoted to a more detailed discussion on solving a multi-dimensional composition-dependent PBE via three specific solution methodologies: a “rigorous” discrete method, the direct quadrature method of moments (DQMOM) technique, and cNMC. The rigorous method is very detailed and accurate, but its accuracy is limited to early granulation times due to its large computational load. Because DQMOM tracks moments of a population rather than its entire distribution, it is a fast algorithm; however, the increased computational speed is gained at the cost of a loss of detailed information about the population. The cNMC technique permits the computation of univariate and multivariate distributions over arbitrarily long times, but is not well-suited for systems that involve space and time gradients. It is found that cNMC is currently the best choice when analyzing multi-dimensional, composition-dependent PBEs.

Chapter 4 deals with the inclusion of a mass-flow stream into the cNMC simulation box. Here, the typical scenario of a continuous addition of binder to the granulation system is analyzed. Of particular interest is the fact that, despite using the same binder type and concentration and similar binder flow rates, experiments show that the individual granulation of three powders produce three very different granulation growth profiles. The interaction of binder with particle morphology was studied via cNMC; it is found that simulations match well with the experimental data for all three growth profiles.

Chapter 5 examines the impact of binder moisture and drying on the granules
produced via fluid bed granulation. During fluid bed granulation, the air used to fluidize the bed also serves to dry the granules. The rate of binder drying directly affects the granulation growth profile. As moisture is removed from a granule via evaporation, the binder may eventually reach a point where it is no longer effective in promoting granulation (i.e., the binder is no longer “sticky” and colliding granules rebound rather than stick together). Industrially, it has been observed that granules with a higher moisture content, as measured by a technique known as LOD (loss on drying), typically promote better granulation. Simulations are presented that qualitatively match expected experimental observation (e.g., a higher temperature causes a decrease in the granule growth rate due to faster drying of the binder) and quantitatively mimic actual experimental results.

Chapter 6 discusses the culmination of this dissertation: how Monte Carlo is used to simulate a multi-dimensional, multi-component granulation. Specifically, a four-component granulation is studied that tracks the evolution of two different powders and both the solid and liquid (moisture) components of the binder. Such a scenario routinely exists within the pharmaceutical industry where a drug compound (powder 1) and an excipient (powder 2) are granulated together with a binder. The impact of compositional effects on granule growth and the potential segregation of powders as a result is examined. The simulation results trend well with the experimental results.

Chapter 7 summarizes the main conclusions derived from this dissertation and provides a status update on the overriding research objective of obtaining a realistic, comprehensive granulation simulator. In addition, this chapter also touches on the future work that may proceed either as a direct result of this research or complementary to it.
Chapter 2

Background

2.1 Chapter Overview

This chapter begins with an overview on the numerical framework used to model granulation: population balance equations (PBEs). A brief description of applications where PBEs are useful is given before a generic coagulation equation is presented. The chapter then moves into a discussion on the need for multi-dimensional and multi-component PBEs and the complexities inherent in the solution of such equations. A key component of any PBE that features coagulation is the choice of what is known as a kernel. A list of some of the kernels encountered in the literature is presented before providing more detail on the kernel often used to describe fluidized beds: the KTGF (kinetic theory of granular flow) kernel. The chapter continues with a brief listing of the various numerical techniques used to solve PBEs before concluding with a more thorough discussion of the basics associated with the solution technique featured throughout this dissertation, constant-number Monte Carlo.

2.2 Population Balance Description

A population balance is nothing more than a number balance on particles with a specific attribute or attributes within a population. More formally, PBEs are a set of integro-differential equations that mathematically describe how this attribute...
(or attributes) of a population evolves over time. PBEs are used in a wide variety of fields that include the study of polymerization [23, 24], crystallization [25, 26], rain drop formation [27, 28], interstellar dust [29, 30], colloids [31, 32], aerosols [33, 34] and others. Regardless of the particular application, the simplest PBEs describe the rate of change in the number density of particles with some attribute $x$. Several processes can change the number of particles within a system, including nucleation, surface growth, reaction, convection into and out of a control volume, dissolution, etc. If we let $F(V_1, t)$ represent the number density function of particles with size $V_1$, then the most general form of a PBE is given by:

$$\frac{\partial F(V_1, t)}{\partial t} = R^+(V_1) - R^-(V_1)$$

(2.1)

where $R^+$ represents the rate per unit volume at which particles with size $V_1$ are formed and $R^-$ represents the rate per unit volume at which particles with size $V_1$ are removed. In a granulator, the applicable rates may include convection, growth via layering, nucleation, aggregation, and breakage. The general PBE for a granulation process is thus written as [4, 35, 36]

$$\frac{\partial F(V_1, t)}{\partial t} = \underbrace{\frac{Q_{\text{in}}}{V} F_1(V_1) - \frac{Q_{\text{out}}}{V} F_1(V_1)}_{\text{convection}} - \underbrace{\frac{\partial [G(V_1, t) F(V_1, t)]}{\partial V_1}}_{\text{growth}} + \underbrace{B_{\text{nuc}}(V_1, t)}_{\text{nucleation}} + \underbrace{B_{\text{agg}}(V_1, t) - D_{\text{agg}}(V_1, t)}_{\text{aggregation}} + \underbrace{B_{\text{break}}(V_1, t) - D_{\text{break}}(V_1, t)}_{\text{breakage}}$$

(2.2)

where $F(V_1)$ is the number density function ($#/\text{kg m}^3$) of particles with total mass $V_1$ (kg), $Q_{\text{in}}$ and $Q_{\text{out}}$ represent the flow rate (m$^3$/s) of particles into and out of the granulator in the case of a continuous process, $V$ corresponds to the granulator volume (m$^3$), $B_{\text{ext}}$ is the birth rate of particles due to a particular process, and $D_{\text{ext}}$ is the death rate of particles due to a particular process. In a closed system such as the Wurster fluidized bed, the convection terms are not applicable. In
addition, the gentle nature of the fluidized bed minimizes the amount of breakage relative to the amount of agglomeration. An assumption is also made that attrition of fine particles from bigger granules does not occur. Thus, we may eliminate the breakage and growth by layering terms. Finally, we assume that nucleation can be considered via the aggregation of binder droplets with primary particles. The resulting equation contains only the aggregation terms. From it, the famous Smoluchowski coagulation equation [37] may be written:

\[
\frac{\partial F(V_1)}{\partial t} = \frac{1}{2} \int_0^{V_2} K(V_1 - V_2, V_2) \ F(V_1 - V_2) \ F(V_2) \ dV_2 - F(V_1) \int_0^\infty K(V_1, V_2) \ F(V_2) \ dV_2,
\]

where \( K(V_1, V_2) \) represents the coagulation kernel (m\(^3\)/s) between two particles. The first term on the right-hand-side of the equation corresponds to the generation of particles of size \( V_1 \) due to the combination of smaller particles whereas the second term describes the depletion of particles of size \( V_1 \) as particles of this size combine with other particles. The factor of \( 1/2 \) is necessary to account for the “double-counting” that occurs when evaluating all particles that can combine to form a particle of size \( V_1 \) (e.g., a granule of size 5 can be formed from a combination of particles of sizes 1 and 4, 2 and 3, 3 and 2, and 4 and 1). Though this equation only describes the evolution of particles based on one parameter, particle mass, it lays the foundation for developing the multi-component PBEs necessary to evaluate pharmaceutical granulation.

### 2.3 Multi-dimensional/Multi-component Population Balance Equations

Population balance models have been applied to granulation quite broadly [38–42]. Traditionally, they treat granules as fully described by a single state variable, their size (see Eq. 2.3). Such models are also referred to as one-dimensional. Nonetheless, the product of granulation cannot be described by a single variable [43].
Composition, porosity, particle morphology, binder viscosity and moisture are important parameters that can affect the final product. As a result, multi-dimensional (also sometimes referred to as multi-component) models that incorporate the effect of such parameters have recently been developed \[44–52\]. The application of multi-dimensional population balance equations (PBEs) to granulation has increased dramatically in recent years.

While many parameters can be considered for multi-dimensional analysis, the key variables often included with size include granule porosity and moisture. As such, 3-D models that track the evolution of solid, liquid, and gas within a granule are often utilized \[51–54\]. Several papers focus on obtaining a quick and accurate numerical solution of the complex PBE \[49,52,55,56\]. However, there also exists a substantial body of work that compares experimental data with simulations based on PB models \[1,35,57–60\]. Ideally, a PB model should accurately represent the physical system it models and compare well with experimental data. Though 3-D models are very attractive, porosity measurements remain difficult to generate and validate \[61\]. Furthermore, for the type of granulation studied in this dissertation (fluidized bed granulation), consolidation, or compression, of the granules is not as prevalent as would be the case in high shear granulation. As such, this work focuses on the consideration of binder (or liquid) content as a key internal property in the development of multi-dimensional population balances. Much research has been devoted to the study of binder content within granules from both an experimental \[62–66\] and a modeling perspective \[49,55,59,61\].

Implicit in the inclusion of moisture as a key granulation variable is the assumption that granule composition plays an important role during granulation. However, this compositional effect is not only restricted to binder content. Another compositional effect to consider is the relative amount of powder in granules that may contain two or more powders. Conceptually, this potential for granule inhomogeneity can be explained by examining two primary powder particles with different surface morphologies: one very rough surface with great asperity and one relatively smooth surface. When liquid binder is applied to the surface of these particles, the particles with large asperity will have binder caught within the crevices and, therefore, less binder will be accessible to promote successful collisions. On the other hand, most of the binder on the smooth particle \textit{will} be readily available.
As such, it is conceivable that the smooth particles will immediately granulate together whereas the rough particles will need to undergo a coating period before being incorporated in the granule. This inhomogeneity has the potential to cause a size-dependent variation in composition and, as a result, a segregation of granules. In the pharmaceutical industry, segregation of multi-component granules continues to be studied [16, 18, 67, 68]. For this reason, powder composition will also be incorporated into the multi-component population balance model developed in this research.

In this dissertation, we focus on a population balance model that describes the granulation of one or two powders with binder. At a minimum, a two-component PB model is utilized (see Chapters 3 and 4). However, three-component and four-component PB models are also studied (see Chapters 5 and 6, respectively). Simply adding just one more dimension to the standard 1-D PBE greatly increases its complexity. An example of a two-component form of the Smoluchowski equation given by Lushnikov [69] is given as:

\[
\frac{\partial F(V_1, U_1)}{\partial t} = \frac{1}{2} \int_0^{V_1} \int_0^{U_1} K(V_1 - V_2, U_1 - U_2; V_2, U_2) \times F(V_1 - V_2, U_1 - U_2) F(V_2, U_2) \, dV_2 dU_2 - F(V_1, U_1) \int_0^{\infty} \int_0^{\infty} K(V_1, U_1; V_2, U_2) F(V_2, U_2) \, dV_2 dU_2
\]

where \( F(V_1, U_1) \) is the number density function (#/kg²m³) of particles with total mass \( V_1 \) (kg) and binder mass \( U_1 \) (kg), \( F(V_1, U_1) dV_1 dU_1 \) represents the number concentration (#/m³) of these particles and \( K(V_1, U_1; V_2, U_2) \) represents the coagulation kernel (m³/s) between two particles. The addition of more components adds additional integrals to each to each term on the right-hand-side of the equation.

### 2.4 Kernel Overview

The physics of the process is governed by the aggregation kernel, \( K_{12} = K(V_1, U_1; V_2, U_2) \), and is a function of the properties of the two colliding particles. The kernel is pri-
marily affected by two factors [10]: (i) the collision probability of the particle pair and (ii) a success factor which specifies the probability that a collision is successful (i.e., that the two particles will remain “stuck” together). The collision probability is mainly influenced by particle size whereas the success factor is strongly influenced by factors such as binder concentration. To accommodate these influences, the kernel is traditionally broken into a size-independent portion (denoted by $K_0$) and a size-dependent portion (the rest of the kernel). The size-independent part may or may not be time-dependent and typically incorporates several system-specific parameters and/or operational conditions. The size-dependent part contains the influence of size on the coagulation rate. A list of some kernels used in the literature to model granulation is given in Table 2.1; $u$ and $v$ represent the size of two different particles.

These kernels range in complexity from simple kernels where an analytical solution of the PBE is possible to kernels that are empirical or semi-empirical in nature to kernels that are more firmly rooted in theory. Analytical solutions exist for only a very few simplified kernels. Examples of kernels with analytical solutions include the first three kernels listed in Table 2.1: the constant, sum, and product kernels [4, 44, 74]. The constant kernel is the simplest form and dictates that particles collide completely at random (i.e., not based on size). While it lacks any physical rationale for its construction, it is often chosen as a starting point when the real kernel is unknown [50]. The sum kernel favors the coagulation of larger particles such that the smallest particles are nearly always present [75]. The product kernel is a variation of the sum kernel where it is predicted that large granules grow even faster (to the point of gelation). The shear kernel also exhibits similar behavior, but is based on the assumption that particles collide due to shear (such as that experienced in a high shear mixer). The other kernel expressions observed in the literature are constructed to cover the vast array of observations seen in industry, ranging from cases where equal-sized particle collisions are favored to the promotion of collisions between granules of dissimilar sizes. In the past, due to the complex interaction of the processing variables affecting granulation, empirical kernels were developed that needed to be fit to laboratory or factory data. Examples include the preferential coalescence of iron ore and/or limestone given in the table. While empirical kernels continue to be developed and honed, increasing em-
Table 2.1: Coalescence Kernels

<table>
<thead>
<tr>
<th>Kernel</th>
<th>Comments</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>$K_{12} = K_0$</td>
<td>Constant kernel</td>
<td>[38]</td>
</tr>
<tr>
<td>$K_{12} = K_0(u + v)$</td>
<td>Sum (Golovin) kernel</td>
<td>[70]</td>
</tr>
<tr>
<td>$K_{12} = K_0(uv)$</td>
<td>Product kernel</td>
<td></td>
</tr>
<tr>
<td>$K_{12} = K_0(u^{1/3} + v^{1/3})^3$</td>
<td>Shear Kernel</td>
<td>[71]</td>
</tr>
<tr>
<td>$K_{12} = K_0 \left( \frac{(u+v)^a}{(uv)^b} \right)$</td>
<td>Preferential coalescence of limestone; $a$ and $b$ are empirical constants</td>
<td>[72]</td>
</tr>
<tr>
<td>$K_{12} = K_0 \left( \frac{u^{2/3} + v^{2/3}}{u^{1/3} + v^{1/3}} \right)$</td>
<td>Preferential balling of iron ore and limestone</td>
<td>[39]</td>
</tr>
<tr>
<td>$K_{12} = K_0 \left( \frac{1}{u^{1/3}} + \frac{1}{v^{1/3}} \right)(u^{1/3} + v^{1/3})$</td>
<td>Brownian kernel</td>
<td>[71]</td>
</tr>
<tr>
<td>$K_{12} = K_0(u^{1/3} + v^{1/3})^2 \sqrt{\frac{1}{u^2} + \frac{1}{v^2}}$</td>
<td>Equipartition of Translational Momentum (ETM) kernel</td>
<td>[73]</td>
</tr>
<tr>
<td>$K_{12} = K_0(u^{1/3} + v^{1/3})^2 \sqrt{\frac{1}{u} + \frac{1}{v}}$</td>
<td>Equipartition of Kinetic Energy (EKE) kernel</td>
<td>[73]</td>
</tr>
</tbody>
</table>

emphasis has been devoted to developing kernels based on first principles. Examples include kernels incorporating the theory that particles collide due to the influence of Brownian motion [76] or random motion associated with the equipartition of translational energy [77] or kinetic energy [78].

### 2.4.1 KTGF Kernel

The equipartition of kinetic energy kernel listed in Table 2.1 has the same form as the kernel associated with the free molecule regime in the kinetic theory of gases
Figure 2.1: Application of collision theory for derivation of size-dependent portion of KTGF kernel.

[76, 79]. This kernel is derived using collision theory which states that collisions between particles are a function of the relative velocity of two particles and their collision cross section. The kernel assumes that kinetic energy is equally distributed across all particles in the system; as such, smaller particles will have a higher velocity relative to larger particles. As two particles approach each other (particles \( A \) and \( B \) in Figure 2.1a) the effective collision cross sectional area is that area through which the center of particle \( A \) must pass in order for a collision to occur. If the radius of particle \( A \) is denoted as \( r_A \) and particle \( B \)'s radius as \( r_B \), then the radius of the collision cross section is \( r_{AB} \) and its area is given as \( \pi r_{AB}^2 \). This cross sectional area is shown schematically in Figure 2.1b. A key characteristic of this kernel is that interactions between large and small particles are favored. Two small particles will move quickly, but have a small “target” area for collision whereas two large particles will have a larger collision cross section, but move more slowly. The most probable scenario occurs when a quickly moving small particle collides with a large particle.

A kernel of similar form can be derived using concepts associated with the ki-
netic theory of granular flow (KTGF) [80, 81]. Key assumptions associated with this theory include the concept that each particle moves freely and independently from its neighbors and experiences a phenomenon known as molecular chaos. In addition, all collisions are binary interactions (i.e., tertiary and higher degree collisions are not considered) due to the speed of the collisions, but these collisions are inelastic and involve energy loss. These assumptions involve some simplification of reality, but provide a good foundation from which to build. In the work of Tan et al. [81], it was shown that the distribution of particle velocities within a fluidized bed are Maxwellian in nature and match that expected from the KTGF. As such, the collision rate between particles can be obtained from this kernel. The interested reader is directed to the work of Tan et al. for a more thorough discussion of the theory surrounding the development of the KTGF kernel, especially in regard to the motion of particles within a continuous granular medium. Because the experimental system under consideration is a fluidized bed, we use a kernel that is strongly rooted in the kinetic theory of granular flow [15] throughout most of this dissertation. In dimensional form, the kernel is given as

$$K_{12} = \psi_{12} \beta d_{43}^\gamma (d_1 + d_2)^2 \left( \frac{1}{d_{1}^4} + \frac{1}{d_{2}^4} \right)^{0.5}. \quad (2.5)$$

Here, $\psi_{12}$ is the dimensionless collision success factor (discussed in more detail in the next section) and is the composition-dependent portion of the kernel, $\beta$ is a coagulation constant (system-specific) with units of $1/(s \ m^{\gamma-2.5})$ and $d_{43}$ is the population volume mean diameter (calculated via the ratio of the fourth to the third moment) with units of meters; the exponent $\gamma$ associated with this parameter incorporates the impact of granular temperature. The remainder of the kernel is the size-dependent portion derived from basic collision theory; $d_i$ is the diameter of a colliding granule with units of meters.

To express the kernel in dimensionless form, we define a dimensionless diameter as $x_i = d_i/d_b$ with a characteristic size, $d_*$, which we choose to be the size of the monodisperse binder drop ($d_b$). Assuming the bulk density of the granules to be constant, we note that the dimensionless mass of a particle, $v_i$, and the
dimensionless diameter are related in a very simple way:

\[ v_i = x_i^3. \]  

(2.6)

The kernel is now written as

\[ K_{12} = K_0 k_{12} \]  

(2.7)

with

\[ K_0 = \beta d_0^{1+0.5} \]  

(2.8)

\[ k_{12} = \psi_{12} x_{13} z_{12} \]  

(2.9)

\[ z_{12} = \left( \frac{v_1^{1/3} + v_2^{1/3}}{v_1 + v_2} \right)^{1/2}. \]  

(2.10)

Here, the \( K_0 \) term is a coagulation rate constant that contains all dimensional parameters and has units of \( \text{m}^3/\text{s} \); \( k_{12} \) contains the size-dependent portion of the kernel. A more thorough discussion on the derivation of the dimensionless form of this KTGF kernel may be found in section 2.6.1.

### 2.4.2 Kernel Success Factor

As mentioned above, collisions alone do not guarantee that colliding particles aggregate. At the most basic level, when two particles collide, they will either rebound or remain stuck together as illustrated by Figure 2.2. Similar to a chemical reaction, other conditions such as proper orientation and impact energy must be considered. If the particles hit each other on a dry spot or the particles are traveling very fast when they collide, rebound is likely. As such, a collision success factor must also be incorporated into the kernel. Throughout most of this dissertation, a success factor that accounts for the wettability and surface coverage of granules by binder [6,15,57,82] based on microscopic models for binder-granule dynamics [2,83] is used. Specifically, we combine the size-dependent KTGF kernel with a collision success factor that accounts for the sticking probability of granules based on both the fractional coverage of the granule surface by the binder and the velocity of the colliding granules.

The success factor \( \psi_{12} \) accounts for the probability of a collision to produce
Figure 2.2: Possible outcomes of a collision. Particle $A$ is red, particle $B$ is purple, and the binder coating is green. Figure is adapted from [1].

The term $\psi_{12}$ is the product of a geometric and physical factor in the form

$$\psi_{12} = \psi_{\text{geom}} \psi_{\text{phys}}.$$  \hfill (2.11)

The geometrical portion considers the probability that two granules will hit each other on a wet surface based on microscopic models for binder-granule dynamics [2, 15, 83]. The physical portion of the success factor takes into account whether or not two colliding particles remain stuck together or rebound by considering the relationship between binder surface coverage, binder viscosity and particle inertia [6, 15, 57, 82].

The geometrical success factor accounts for the accessible binder fraction on a granule and is given by the equation [15]:

$$\psi_{\text{geom}} = 1 - (1 - \eta_1)(1 - \eta_2).$$  \hfill (2.12)

The accessible fraction of binder is the fraction of a granule’s surface area that is covered by binder and available for coagulation; it not only depends on the amount of binder, but also the surface roughness and wettability of the powder. Stepanek et al. provide the following correlation for this quantity [2]:

$$\eta_i = \frac{1}{1 + e^{-b(y_i - c)}},$$  \hfill (2.13)
where $y_i$ is the binder to solid ratio of particle $i$, while $b$ and $c$ are empirical correlation parameters obtained by Stepanek et al. [2] and listed in Table 2.2 for the three powders used throughout the majority of this dissertation (see chapters 4, 5, and 6). Optical images of the powders are shown in Figure 2.3. Avicel has the smoothest surface of the three, whereas A-TAB has the roughest. This relation is especially reflected in the values of parameter $c$; Avicel has the smallest $c$ value and A-TAB has the largest. Higher values of $c$ effectively serve to decrease $\eta$ for a given binder to solid ratio due to a need to fill in the crevices before binder can be used for coagulation.

The physical success factor takes into account a Stokes number analysis to
Table 2.2: Particle shape descriptors for experimental excipients

<table>
<thead>
<tr>
<th>Excipient</th>
<th>b</th>
<th>c</th>
<th>f</th>
<th>g</th>
<th>h_a</th>
</tr>
</thead>
<tbody>
<tr>
<td>Avicel</td>
<td>35.1</td>
<td>0.129</td>
<td>25.9</td>
<td>0.327</td>
<td>6.3</td>
</tr>
<tr>
<td>Mannitol</td>
<td>27.4</td>
<td>0.182</td>
<td>26.8</td>
<td>0.394</td>
<td>10.5</td>
</tr>
<tr>
<td>A-TAB</td>
<td>33.5</td>
<td>0.333</td>
<td>30.4</td>
<td>0.538</td>
<td>16.5</td>
</tr>
</tbody>
</table>

determine the potential success of a collision \([3, 6, 82, 84]\). A collision is successful if the Stokes number of colliding particles remains below a critical value:

\[
\psi_{\text{phys}} = \begin{cases} 
1 & \text{if } St \leq St^* \\
0 & \text{if } St > St^* 
\end{cases}
\] (2.14)

It accounts for collisions of Type I and Type II, where the former refers to sticking that occurs before the particle surfaces come into contact, and the latter to sticking that takes place after the particle surfaces collide and begin to rebound. The Stokes number of colliding particles is \([82]\)

\[
St = \frac{4 \rho u_0 d_1 d_2 (d_1 + d_2)^2}{9 \mu (d_1^3 + d_2^3)}
\] (2.15)

where \(\rho\) is the powder density (kg/m\(^3\)), \(u_0\) is the velocity of the collision (m/s), \(\mu\) is the binder viscosity (kg/(m s)), and \(d_i\) is the diameter (m) of colliding particle \(i\). A thorough discussion on the derivation of the Stokes number may be found in section 2.6.2.

The critical Stokes number is calculated as \([15, 82]\)

\[
St^* = 2 \ln \frac{\lambda_{12}}{h_a}
\] (2.16)

and depends on the binder layer thickness between the two colliding particles, \(\lambda_{12}\), and the asperity (roughness) of the primary particle surface, \(h_a\). The binder layer thickness is calculated from the relative displaced binder volume \(\phi_i\) as per the relationship \([2, 15]\)

\[
\lambda_{12} = r_g \phi_1^{1/3} + r_g \phi_2^{1/3}
\] (2.17)
where \( r_g \) is the mean radius of gyration. A constant value of \( r_g = 60 \, \mu m \) is used for all simulations; this value is the size of the primary particle used in the creation of the “virtual granules” from which the empirical correlation parameters contained in \( \phi_i \) are derived. The relative displaced binder volume \( \phi_i \) corresponds to the volume of binder present on the surface of granule \( i \) available for coagulation. It is a function of the binder to solid ratio, \( y_i \), and given by the equation

\[
\phi_i = e^{f(y_i - g)}
\]  
(2.18)

where \( f \) and \( g \) are empirical correlation parameters again given by Stepanek et al. [2]. These values, as well as \( h_a \), are listed in Table 2.2 for each of the three powders utilized in this study.

### 2.5 Population Balance Solution Methods

As mentioned in section 2.4, very few analytical solutions to the set of population balance equations exist due to the complexity of the coagulation kernel. In general, one must resort to numerical techniques to solve the equations. Though the numerical solution of a set of PBEs is not trivial, it can be accomplished in a number of ways that vary in their algorithmic details. The solution methodologies can broadly be classified as either deterministic or stochastic in nature. Typical deterministic solution techniques include sectional methods [85–89] and the method of moments [90–98]. Stochastic models utilize Monte Carlo to track the evolution of a population’s characteristics [61,99–108].

#### 2.5.1 Sectionalization Techniques

Sectional methods, also known as discretization methods, approximate the continuous size distribution by subdividing it into discrete sections. As such, in the case of a one-dimensional population balance where only particle size is considered, a set of ordinary differential equations of the following form is written for each section [109]:

\[
\frac{dN_i}{dt} = f(N_1, N_2, \ldots, N_M)
\]  
(2.19)
where $N_i$ is the number of particles in section $i$ and $M$ is the total number of sections. The judicious selection of the discretization ranges is a key step in this numerical technique. While a linear subdivision appears to be a logical choice, much information about the smaller particles in the system is lost unless a very large number of classes is utilized. Another technique is to subdivide the particle size domain into a geometric series such that $v_{i+1}/v_i = 2$ \cite{110}. This method strikes a balance between keeping the number of subdivisions low while still maintaining accuracy in the smaller size ranges. One drawback of this method stems from the fact that the desired resolution of the final product is not known a priori. Specific features of the distribution have the potential to become lost in the discretization. Another drawback is encountered when the solution of PBEs with more than one dimension is desired. In this case, an equation is necessary for each subset of the particle domain under consideration. For instance, if 50 classes are desired to evaluate size and 50 classes are desired to evaluate liquid content, then $50 \times 50 = 2,500$ equations are necessary. Tracking a third dimension with the same level of resolution requires 125,000 equations. As can be seen, increasing the dimensionality of a problem drastically increases the computational load required to solve the problem. However, once all equations are formulated, the set of ODEs may be solved via any standard ODE solver.

### 2.5.2 Method of Moments

The method of moments is a numerical technique that simplifies the population balance equation by considering only the evolution of a population’s moments rather than detailed information about the entire population distribution. This reduction in the level of detail required greatly reduces the computational burden and results in a very efficient calculation. A brief tutorial on the general concept of the method of moments is outlined below.

When only size is considered, the PBE as given by the Smoluchowski equation (see Eq. 2.3) represents the change of one particle size within a population. To determine the rate of change in the population number concentration, one needs
to integrate the PBE with respect to $V_1$ over the entire population:

$$
\frac{\partial}{\partial t} \int_0^\infty F(V_1) \, dV_1 = \frac{1}{2} \int_0^\infty \int_0^\infty K(V_1 - V_2, V_2) \, F(V_1 - V_2) \, F(V_2) \, dV_2 \, dV_1 \\
- \int_0^\infty \int_0^\infty K(V_1, V_2) \, F(V_1) \, F(V_2) \, dV_2 \, dV_1.
$$

(2.20)

Upon recognizing that $\int_0^\infty F(V_1) \, dV_1 = C_p$, where $C_p$ is the total number concentration of particles, the equation may be rewritten as (after simplification):

$$
\frac{dC_p}{dt} = \frac{1}{2} \int_0^\infty \int_0^\infty K(V_1, V_2) \, F(V_1) \, F(V_2) \, dV_2 \, dV_1
$$

(2.21)

For clarity, a constant kernel of $K(V_1, V_2) = 1$ is assumed and this equation may be further simplified to:

$$
\frac{dC_p}{dt} = \frac{1}{2} C_p^2.
$$

(2.22)

Here, a very brief description of moments is necessary.

Using the notation developed above, a moment of order $k$ for a distribution is defined as:

$$
M_k = \int_0^\infty V_1^k F(V_1) \, dV_1.
$$

(2.23)

As such, $M_0 = \int_0^\infty F(V_1) \, dV_1$ which was already given as $C_p$. Thus, the set of PBEs that describe the evolution of a population is transformed into an ODE for moment $M_0$:

$$
\frac{dM_0}{dt} = \frac{1}{2} M_0^2.
$$

(2.24)

Similar differential equations for moments of any order $k$ may be obtained by multiplying the Smoluchowski equation by $V_1^k \, dV_1$ and then integrating over all values for $V_1$. For further details on this procedure, the reader is referred to an excellent paper by Michael Frenklach [111]. This solution technique is very efficient, but its gain in computational speed is offset by a loss of detail about the population.
Thus, the method of moments is most useful when information about the entire particle size distribution is not needed and quantities that can be calculated via moments (e.g., mean, variance, skewness, and kurtosis) are sufficient.

### 2.5.3 Stochastic Techniques

Monte Carlo simulations stochastically track a discrete subset of particles, typically around $10^4$, within a simulation box as they undergo size transformations by coagulation or other processes. The behavior of these particles during the simulation is indicative of the behavior of the entire population. Particle attributes are captured in individual arrays that can contain any number of intrinsic properties, including but not limited to granule and binder mass. As such, this numerical solution technique is a logical choice when considering multivariate problems. The main disadvantage of Monte Carlo is that it is not easily interfaced with standard process simulators and so cannot be coupled with hydrodynamic simulators such as FLUENT.

Two general classifications of Monte Carlo exist. One approach is known as time-driven Monte Carlo; in this technique, a time step $\Delta t$ is first specified and then Monte Carlo is used to determine the number and type of events that occur within this time [112, 113]. The other approach is event-driven Monte Carlo whereby an event is first selected and implemented and then time is advanced accordingly as per the rate of the selected process [114, 115]. While time-driven and event-driven methodologies both calculate the size distribution with similar accuracy, the time-driven methodology tends toward larger error when determining the number and mass distributions [102]. For this reason, this dissertation focuses on event-driven Monte Carlo simulations.

Within the event-driven Monte Carlo classification, a further subdivision can be made: constant-volume Monte Carlo and constant-number Monte Carlo. During constant-volume Monte Carlo, the simulation box is treated in a similar manner to that observed in the granulator as a whole; as each coagulation event occurs, the total number of particles in the simulation box decreases by one. This simulation is obviously limited to only having $N$ coagulation events where $N$ is the initial number of particles. One way to circumvent this problem is to conduct the simulation until...
half the original number of particles remain in the simulation box. At this point, all particles are duplicated and the simulation continues. The doubling process does not impact the size distribution and permits the simulation to proceed indefinitely. A natural extension of this technique is constant-number Monte Carlo in which a randomly selected particle is replicated after each coagulation event such that a constant number of particles is maintained in the system. This technique also permits the simulation of long growth times with a finite number of simulation particles and is the methodology used throughout this entire dissertation.

2.5.4 Development of Constant-Number Monte Carlo

In Monte Carlo, each process that affects the particle distribution is simulated as a stochastic event. At a minimum, the process of coagulation is considered, but other processes that may be considered include the introduction of binder droplets, evaporation of moisture, and many others. The basic outline of the algorithm is as follows:

1. Calculate the probability of each event.
2. Choose the event to occur via a uniform random number generator.
3. Implement the selected event. For agglomeration, two particles are selected and combined into a new granule. A randomly selected particle is then duplicated and added into the simulation box to keep the number constant.
4. Update all metrics (mass concentration, number concentration, time advancement, etc.)

Because cNMC is an event-driven process, only one event is implemented at each step of the simulation. The selection of event \( l \) is made with probability \( P_l \) based on the rate of that event

\[
P_l = \frac{R_l}{\sum_i R_i}
\]  

(2.25)

where \( R_i \) is the rate per unit volume (s\(^{-1}\) m\(^{-3}\)) of event \( i \) and the summation in the denominator runs through all possible events.
The dimensionless coagulation rate is \[99\]

\[ r_{\text{coag}} = \frac{\langle k_{12} \rangle c_p^2}{2 \tau_{\text{coag}}/t_*} \] (2.26)

where \(\langle k_{12} \rangle\) is the mean dimensionless coagulation kernel between all particle pairs, \(c_p\) is the dimensionless number concentration of particles defined as \(c_p = C_p/C_*\), \(t_*\) is a characteristic time (to be specified in section 2.5.4.3) and \(\tau_{\text{coag}}\) is the coagulation time, defined as

\[ \tau_{\text{coag}} = \frac{1}{K_0 C_*}. \] (2.27)

The value of the characteristic concentration, \(C_*\), will be specified in section 2.5.4.3. A more thorough discussion on the derivation of the dimensionless form of the coagulation rate may be found in section 2.6.3.

Events are implemented by the general constant-number algorithm \([99–101]\). For coagulation, a pair of granules is elected with probability proportional to the dimensionless kernel \(k_{ij}\) between them. Upon coagulation, a new granule is formed whose constituent components (solid excipient and binder) is the sum of those in the parent granules. The constant number of simulation particles is enforced by duplicating a granule at random and placing the copy in the simulation box.

### 2.5.4.1 Concentrations

Number and mass concentration are tracked during simulation and updated accordingly following the implementation of an event. The derivation of these expressions and their validation has been given previously \([101, 102, 107]\). Here, we summarize the main expressions needed in the calculation. The dimensionless mass concentration, \(c_m\), is calculated as follows:

\[ c_m = c_{m,\text{old}} \left( 1 + \frac{\delta v_{\text{event}}}{N \bar{v}} \right). \] (2.28)

Here, \(c_{m,\text{old}}\) is the dimensionless mass concentration from the previous Monte Carlo step, \(N\) is the number of simulation particles, \(\bar{v}\) is the mean dimensionless particle mass in the simulation box, and \(\delta v_{\text{event}}\) is the change in the mass of the simulation box as a result of the event chosen. For coagulation, this calculation is trivial as
\( \delta \nu_{\text{event}} = 0 \) (no net change of mass). The dimensionless number concentration is calculated as

\[
c_{p} = \frac{c_m}{\bar{v}},
\]

(2.29)

where \( c_m \) is the mass concentration calculated above and \( \bar{v} \) is as defined previously.

2.5.4.2 Time Increment

When only coagulation is considered, the mean dimensionless time that elapses between Monte Carlo events is given by [101]:

\[
\Delta \tau = \frac{c_p}{N r_{\text{coag}}}
\]

(2.30)

and is calculated based on the current values of the number concentration \( c_p \) and event rate \( r_{\text{coag}} \). A more thorough discussion on the derivation of the dimensionless time increment may be found in section 2.6.4.

2.5.4.3 Characteristic Dimensional Variables

The quantities used to formulate the problem in dimensionless form may be chosen arbitrarily. For the calculations shown here, the following characteristic variables are used:

- time: \( t_{\ast} = \tau_{\text{coag}} \) (s)
- concentration: \( C_{\ast} = C_0 \) (m\(^{-3}\))
- volume: \( V_{\ast} = 1/C_{\ast} \) (m\(^3\))
- size: \( d_{\ast} = d_b \) (m)

where \( \tau_{\text{coag}} \) is the coagulation time, \( C_0 \) is the initial number concentration of granules, and \( d_b \) is the monodisperse binder droplet diameter. With these values now specified, the important equations necessary to conduct constant-number Monte Carlo may be finalized; a summary of these equations is given in Table 2.3. While several of these equations are trivial when only coagulation is considered, the rigorous development presented in this section gives the framework for the calculation of these parameters when more complex scenarios are considered (e.g., when binder flow into the system and evaporation are considered in Chapters 4 and 5, respectively).
Table 2.3: Key constant-number Monte Carlo equations for coagulation

<table>
<thead>
<tr>
<th>Quantity</th>
<th>Equation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rate</td>
<td>( r_{\text{coag}} = \frac{\langle k_{12} \rangle c_p}{2} )</td>
</tr>
<tr>
<td>( \tau_{\text{event}} )</td>
<td>( \tau_{\text{coag}} = \frac{1}{K_0 C_0} )</td>
</tr>
<tr>
<td>Probability</td>
<td>( P_{\text{coag}} = \frac{r_{\text{coag}}}{r_{\text{coag}}} = 1 )</td>
</tr>
<tr>
<td>( \delta v_{\text{event}} )</td>
<td>0</td>
</tr>
<tr>
<td>Mass concentration</td>
<td>( c_m = c_{m,\text{old}} \left( 1 + \frac{\delta v_{\text{event}}}{N_e} \right) )</td>
</tr>
<tr>
<td>Number concentration</td>
<td>( c_p = \frac{c_m}{\bar{v}} )</td>
</tr>
<tr>
<td>Time between events</td>
<td>( \Delta \tau = \frac{c_p}{N} \frac{1}{r_{\text{coag}}} )</td>
</tr>
</tbody>
</table>

2.6 Derivation of Equations

2.6.1 Dimensionless KTGF Kernel Derivation

The modified KTGF kernel is given as:

\[
K_{12} = \psi_{12} \beta d_{13}(d_1 + d_2)^2 \left( \frac{1}{d_1^3} + \frac{1}{d_2^3} \right)^{0.5}
\]  

(2.31)

where all parameters are as given before. To make the particle size dimensionless, we choose as characteristic size the initial monodisperse binder droplet diameter, \( d_b \). In this manner, all binder droplets will have a dimensionless size of 1 and all granule sizes will be relative to the binder droplet size. We thus define a dimensionless diameter as \( x = d/d_b \) from which it follows that

\[
d = xd_b
\]

(2.32)
The volume of a particle of size \( d \) is

\[
V = \frac{\pi d^3}{6} = \frac{\pi d_b^3}{6} x^3 = V_b x^3
\]  

(2.33)

where the characteristic volume \( V_b \) is the volume of a binder droplet with diameter \( d_b \). Because the mass and volume of a particle are related simply by the particle’s density (mass = volume \( \times \) density) and we assume constant density, we can define the dimensionless mass \( v \) as

\[
v = \frac{V}{V_b}
\]

(2.34)

and note that the dimensionless mass and the dimensionless diameter are related in a very simple way:

\[
v = x^3
\]

(2.35)

Using these dimensionless parameters, Eq. (2.31) may be rewritten as

\[
K_{12} = \psi_{12}\beta d_{43}^0 d_b^2 (x_1 + x_2)^2 d_b^{-1.5} \left( \frac{1}{x_1^3} + \frac{1}{x_2^3} \right)^{0.5} = \psi_{12}\beta d_{43}^0 d_b^{0.5} (v_1^{1/3} + v_2^{1/3})^2 \left( \frac{1}{v_1} + \frac{1}{v_2} \right)^{0.5}
\]

(2.36)

The 43 diameter is defined as

\[
d_{43} = \frac{m_4}{m_3}
\]

(2.37)

where \( m_k \) is the \( k \)-order moment of the diameter. If the population is given as a sample of \( N \) discrete particles, the moment of order \( k \) is

\[
m_k = \sum_i^N d_i^k
\]

(2.38)

Accordingly, the 43 diameter is

\[
d_{43} = \frac{\sum_i^N d_i^4}{\sum_i^N d_i^3}
\]

(2.39)

Using the dimensionless diameter defined in Eq. (2.32), the 43 diameter may be redefined as

\[
d_{43} = d_b \frac{\sum_i^N x_i^4}{\sum_i^N x_i^3} = d_b \frac{\sum_i^N v_i^{4/3}}{\sum_i^N v_i} = \frac{d_b}{M} \sum_i^N v_i^{4/3}
\]

(2.40)
where \( M = \sum_i v_i \) is the total mass (which is not constant in constant number Monte Carlo). The dimensionless 43 diameter, defined as \( x_{43} = d_{43}/d_b \), is

\[
x_{43} = \frac{\sum_i x_i^4}{\sum_i x_i^3} = \frac{\sum_i v_i^{4/3}}{\sum_i v_i} = \frac{\sum_i v_i^{4/3}}{M}
\]

Any of the expressions on the right-hand side of the equation can be used to calculate \( x_{43} \). Using the dimensionless diameters from \( d_i = x_id_b \) and \( d_{43} = x_{43}d_b \), Eq. (2.36) becomes

\[
K_{12} = \psi_{12}\beta x_{43}^\gamma d_b^{0.5}(v_1^{1/3} + v_2^{1/3})^2 \left( \frac{1}{v_1} + \frac{1}{v_2} \right)^{0.5}
\]

which may be recast as

\[
K_{12} = K_0 k_{12}
\]

with

\[
K_0 = \beta d_b^{0.5}
\]

\[
k_{12} = \psi_{12} x_{43}^\gamma z_{12}
\]

\[
z_{12} = (v_1^{1/3} + v_2^{1/3})^2 \left( \frac{1}{v_1} + \frac{1}{v_2} \right)^{1/2}.
\]

The \( K_0 \) term contains all dimensional parameters and has units of \( \text{m}^3/\text{s} \). All other terms in \( K_{12} \) are dimensionless.

### 2.6.2 Dimensionless Stokes Number Derivation

The Stokes number is defined as

\[
\text{St} = \frac{8\tilde{m}u_0}{3\pi \mu \tilde{d}^2}
\]

where \( \tilde{m} \) and \( \tilde{d} \) are the reduced particle mass and diameter, \( u_0 \) is the velocity of the collision, and \( \mu \) is the binder viscosity. More specifically, the reduced particle mass and diameter are

\[
\tilde{m} = \frac{m_1m_2}{m_1 + m_2}
\]
and

\[ \bar{d} = \frac{d_1 d_2}{d_1 + d_2}, \quad (2.49) \]

respectively. The subscripts 1 and 2 refer to the arbitrary labeling of two colliding granules as granule 1 and granule 2. Upon substituting Equations (5.55) and (5.56), along with the relationship \( m = \rho \frac{2d^3}{6} \), and rearranging, the Stokes equation is rewritten as:

\[ \text{St} = \frac{4}{9} \frac{\rho u_0 d_1 d_2 (d_1 + d_2)^2}{\left( \frac{d_1^3}{d_1^3} + \frac{d_2^3}{d_2^3} \right)} \quad (2.50) \]

Using dimensionless granule size, the Stokes number is

\[ \text{St} = \frac{4}{9} \frac{\rho u_0 d_0 \frac{1}{3} v_1^{1/3} v_2^{1/3} (v_1^{1/3} + v_2^{1/3})^2}{(v_1 + v_2)} \quad (2.51) \]

### 2.6.3 Dimensionless Coagulation Rate Derivation

The coagulation rate is derived from the population balance equation defined in Eq. (2.4). When integrating across all masses, \( \int \int F(V_1, U_1) dV_1 dU_1 \) becomes the total number concentration, \( C_p \), of particles in the system. Thus, the L.H.S of Eq. (2.4) is transformed from

\[ \int \int \frac{dF(V_1, U_1)}{dt} dV_1 dU_1 \]

to

\[ \frac{dC_p}{dt}, \]

the rate of change in the number concentration as a function of time. The specific rate (rate per unit volume) of coagulation is then obtained after integrating the R.H.S. of the equation across all masses. After consolidation, the specific coagulation rate is given by:

\[ R_{\text{coag}} = \frac{1}{2} \int \int \int \int K_{12} F(V_1, U_1) F(V_2, U_2) dV_1 dV_2 dU_1 dU_2 \quad (2.52) \]
where $F(V,U)dVdU$ is the number concentration of particles of total mass $V$ and binder mass $U$. If we define the following relationship

$$\langle K_{12} \rangle = \frac{\int \int \int \int K_{12} F(V_1, U_1) F(V_2, U_2) dV_1 dV_2 dU_1 dU_2}{\int \int \int \int F(V_1, U_1) F(V_2, U_2) dV_1 dV_2 dU_1 dU_2}$$

(2.53)

$$= \frac{\int \int \int \int K_{12} F(V_1, U_1) F(V_2, U_2) dV_1 dV_2 dU_1 dU_2}{C_p^2}$$

then the specific rate takes the following form:

$$R_{coag} = \langle K_{12} \rangle \frac{C_p^2}{2}$$

(2.54)

We convert these equations into dimensionless form to simplify analysis. To convert the specific rate, $R_i$ (m$^{-3}$ s$^{-1}$), into a dimensionless rate, $r_i$, we must multiply the specific rate by a characteristic time, $t_*$ (s), and volume, $V_*$ (m$^3$). (As an alternative, we may also multiply the specific rate by a characteristic time, $t_*$ (s), and divide by a characteristic concentration, $C_*$ (1/m$^3$)). The values for $t_*$ and $V_*$ were already defined in section 2.5.4.3 as $\tau_{coag}$ and $1/C_0$, respectively. The dimensionless coagulation rate is thus:

$$r_{coag} = R_{coag} \frac{\tau_{coag}}{C_0} = \frac{\langle K_{12} \rangle C_p^2}{2} \frac{\tau_{coag}}{C_0}$$

(2.55)

We can modify this equation by defining a dimensionless concentration, $c_p = C_p/C_0$, where $C_0$ is the initial particle number concentration and expressing the kernel as

$$K_{12} = \psi_{12} K_0 x_4^\gamma z_{12}.$$  

With these definitions, the dimensionless coagulation rate becomes

$$r_{coag} = \frac{K_0 x_4^\gamma \langle \psi_{12} z_{12} \rangle c_p^2}{2} \frac{\tau_{coag}}{C_0} = (K_0 x_4^\gamma C_0 \tau_{coag}) \frac{\langle \psi_{12} z_{12} \rangle c_p^2}{2}$$

(2.56)
The characteristic time $\tau_{\text{coag}}$ was previously defined as

$$\tau_{\text{coag}} = \frac{1}{K_0 C_0}$$  \hspace{1cm} (2.57)

with units of seconds. Using this definition, the final form of the dimensionless coagulation rate becomes

$$r_{\text{coag}} = x_{43} \frac{\langle \psi_{12} z_{i2} \rangle c_p^2}{2}$$  \hspace{1cm} (2.58)

which is equivalent to

$$r_{\text{coag}} = \frac{\langle k_{12} \rangle c_p^2}{2}$$  \hspace{1cm} (2.59)

given in Table 2.3

### 2.6.4 Dimensionless Time Increment Derivation

The mean time that elapses between Monte Carlo events is given by

$$\Delta t = \frac{1}{V_{MC} \sum R_i}$$  \hspace{1cm} (2.60)

where $R_i$ is the specific rate ($\text{m}^{-3}\text{s}^{-1}$) of a particular process. In constant-number Monte Carlo, the volume of the simulation box, $V_{MC}$, is constantly changing so that, at each successive time step in the Monte Carlo process, the same number of particles is evaluated. Monte Carlo uses a sample of $N$ simulation particles to represent a number concentration $C_p$ distributed over a fluid volume $V_{MC}$. Thus, the real volume that corresponds to the sample in the simulation box is

$$V_{MC} = \frac{N}{C_p}$$  \hspace{1cm} (2.61)

Substituting this relationship into Eq. (2.60), and inserting the rate of coagulation for the sum of rates term, yields

$$\Delta t = \frac{C_p}{N R_{\text{coag}}} \frac{1}{V_{MC}}$$  \hspace{1cm} (2.62)

As before, for efficient analysis, it is convenient to work in dimensionless units.
Using the following relationships

\[ c_p = \frac{C_p}{C_0} \]

\[ r_{\text{coag}} = R_{\text{coag}} \left( \frac{T_{\text{coag}}}{C_0} \right) \]

we obtain

\[ \Delta t = c_p C_0 \frac{1}{N \ r_{\text{coag}} C_0 / (T_{\text{coag}})} = c_p T_{\text{coag}} \frac{1}{N \ r_{\text{coag}}} \]  \hspace{1cm} (2.63)

Plugging the definition of \( r_{\text{coag}} \) from Eq. (2.59) into Eq. (2.63) gives a time advancement of

\[ \Delta t = c_p T_{\text{coag}} \frac{1}{N \langle k_{12} \rangle c_p^2 / 2} \]  \hspace{1cm} (2.64)

Dividing both sides of the equation by \( T_{\text{coag}} \) provides the final derivation for the advancement of dimensionless time in Monte Carlo (when only coagulation is considered):

\[ \Delta \tau = c_p \frac{1}{N \langle k_{12} \rangle c_p^2 / 2} \]  \hspace{1cm} (2.65)
Chapter 3

Numerical Simulations of Two-Component Granulation: Comparison of Three Methods

3.1 Chapter Overview

In this chapter, the impact of composition on the solution of a multi-component population balance equation (PBE) is studied. The system under review is a two-powder granulation that is considered to have binder already present on the surface of the particles; this scenario is similar to the melt in situ conditions encountered in industry. Previous work in Dr. Matsoukas’s research group has focused on the development of a theory of aggregative mixing of components and in the scaling behavior of the compositional distribution \([50, 106, 116]\). The current work aims at the computational tools that can be used to solve the bivariate population balance. Specifically, we consider three methodologies:

(i) Direct solution of the discrete, two-component PBE. This rigorous and detailed approach is computationally demanding and can only be used to track the very early stages of granulation \([15]\).

(ii) Constant-Number Monte Carlo (cNMC). This method allows the computation of univariate and multivariate distributions over arbitrarily long times...
Its chief disadvantage is that it is not well-suited for systems that involve space and time gradients.

(iii) Direct Quadrature Method of Moments (DQMOM): this methodology is very efficient in one-component systems and is currently the only viable option for interfacing the PBE with fluid dynamics [90, 97, 118, 119]. Extension to two-component systems is not straightforward and requires validation against known solutions.

The kernels used in this study stem from the constant kernel and a variation of the kinetic theory of granular flow (KTGF) kernel. A collision success factor equal to 1 (i.e., all collisions result in successful coagulation) is utilized so that the only factor impacting growth and mixing is the collision rate. To evaluate compositional effects, a composition-dependent factor is incorporated into the kernel; this term can easily be modified to reflect a self-aggregating system (i.e., a powder prefers to combine with itself rather than mixing with the second powder), a cross-aggregating system (i.e., one powder prefers to combine with the other powder rather than itself), or some scenario in between. A parameter known as the Content Heterogeneity Index (CHI) is introduced to quantify the internal blending of the two powders. Ultimately, it is found that DQMOM is very accurate when studying systems with no compositional dependence, but the accuracy decreases when compositional-dependence is introduced. Both the rigorous discrete method and constant-number Monte Carlo are accurate with or without compositional-dependence, but the rigorous discrete method is limited to only the very early stages of granulation.

3.2 Multi-component Granulation and Composition

Modeling multi-component granulation has received less attention in the literature than one-component systems, in part due to the fundamental questions in one-component systems that remain unresolved. Nonetheless, even the simplest granulation problem involves at least two components: the solid powder and the binder. Moreover, in most applications this powder is not pure but a segregated mixture
of components. Thus, one needs to know not only the rate of granule enlargement, but also the compositional distribution of components. There are various reasons that lead to inhomogeneous distribution of components within granules. Imperfect mixing can arise from the fluid patterns within the granulator [120, 121]. This is more of an issue in high-shear granulation and less so in bottom spray fluidization, such as the Wurster granulator, which allows for better circulation and contact between the phases. Recent results of detailed simulations employing a model combining CFD (computational fluid dynamics) and population balancing suggest that the aggressive flow patterns in the Wurster unit ensure continuous circulation of material and an absence of dead zones [15]. These observations support the suitability of the well-mixed assumption in the Wurster granulator.

Another important consideration that impacts granule composition is the ability of the binder to effectively coat the granules. Thus, interfacial energy, local curvature and surface roughness of the granules can affect the process in critical ways. In granulation of a single solid, such properties affect the rate of granule growth, and possibly the size distribution, if the morphology of granules and granule/binder interactions vary with size. In granulation of two or more components, such interactions have the additional effect of potentially producing a severely inhomogeneous distribution of components, resulting in poor mixing and even segregation. Similar reports with other systems confirm the view that granulation can produce fairly inhomogeneous distributions of components, manifested as a systematic variation of composition with granule size [16, 17, 122]. Because of the large impact composition can have on the final granules that are produced, it is necessary to have modeling tools that can effectively replicate the conditions that may be seen in industry.

### 3.3 Theoretical Background

The bicomponent population of our study consists of a component of interest (“solute”) and an inert component (“excipient”). We describe this population through a bivariate state vector \((v, m)\) where \(v\) is the mass (or “size”) of a granule and \(m\) is the amount (mass) of solute in the granule. Alternatively, the mass of solute may be replaced by the mass fraction, \(c = m/v\), of solute in the granule.
The bivariate distribution is given by

\[ F(v, c) = G(c|v)f(v) \]  \hspace{1cm} (3.1)

where \( f(v) \) is the usual size distribution and \( G(c|v) \) is the compositional distribution of granules of size \( v \), namely, the probability to find a granule that contains mass fraction of solute \( c \) given that the mass of the granule is \( v \).

The overall mass fraction of solute in the population is

\[ \phi = \frac{M_S}{M} \]  \hspace{1cm} (3.2)

where \( M_S \) is the mass of solute in the population, and \( M \) is the total mass of granules. The population is completely mixed if \( c = \phi \) in all granules. Accordingly, a measure of content heterogeneity is the quantity

\[ \text{CHI} = \frac{\langle v^2(c - \phi)^2 \rangle}{\bar{v}^2} \]  \hspace{1cm} (3.3)

The numerator is the variance of “excess solute,” defined as \( x = v(c - \phi) \), which gives the mass of solute in a granule in excess of the desired amount \( \phi v \); the denominator is the square of the mean granule mass. Their ratio defines the Content Heterogeneity Index (CHI) as a measure of internal blending of the two components. For a well-blended system, \( \text{CHI} \to 0 \).

### 3.4 Moments

We define the moment of mixed order \( i, j \), as

\[ M_{ij} = \int \int (cv)^i(v - cv)^j F(v, c) dv dc \]  \hspace{1cm} (3.4)

in which the mass of solute \( (cv) \) and excipient \( (v - cv) \) are treated symmetrically. The mean granule mass, the fraction of solute, and the variance of excess solute can be easily expressed in terms of mixed moments:

\[ \bar{v} = \frac{M_{10} + M_{01}}{M_{00}} \]  \hspace{1cm} (3.5)
Table 3.1: Summary of cases

<table>
<thead>
<tr>
<th>Case</th>
<th>Size dependence</th>
<th>Comp. dependence</th>
<th>( \phi )</th>
<th>Initial sizes</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Constant kernel</td>
<td>( \kappa = 1 )</td>
<td>0.0909</td>
<td>( v_A = 1, v_B = 10 )</td>
</tr>
<tr>
<td>2</td>
<td>KTGF kernel†</td>
<td>( \kappa = 1 )</td>
<td>0.0909</td>
<td>( v_A = 1, v_B = 10 )</td>
</tr>
<tr>
<td>3</td>
<td>Constant kernel</td>
<td>( \kappa(c_1, c_2)^$ )</td>
<td>0.0909</td>
<td>( v_A = 1, v_B = 10 )</td>
</tr>
<tr>
<td>4</td>
<td>KTGF kernel†</td>
<td>( \kappa(c_1, c_2)^$ )</td>
<td>0.0909</td>
<td>( v_A = 1, v_B = 10 )</td>
</tr>
<tr>
<td>5</td>
<td>KTGF kernel†</td>
<td>( \kappa(c_1, c_2)^$ )</td>
<td>0.0909</td>
<td>( v_A = 2, v_B = 20 ) (normal)</td>
</tr>
<tr>
<td>6a</td>
<td>KTGF kernel‡</td>
<td>( \kappa(c_1, c_2)^$ )</td>
<td>0.0909</td>
<td>( v_A = 2, v_B = 20 ) (normal)</td>
</tr>
<tr>
<td>6b</td>
<td>KTGF kernel‡</td>
<td>( \kappa(c_1, c_2)^$ )</td>
<td>0.0814</td>
<td>( v_A = 2, v_B = 23 ) (lognormal)</td>
</tr>
<tr>
<td>6c</td>
<td>KTGF kernel‡</td>
<td>( \kappa(c_1, c_2)^$ )</td>
<td>0.0814</td>
<td>( v_A = 2, v_B = 23 ) (lognormal)</td>
</tr>
</tbody>
</table>

\( \uparrow \gamma = 0; \)
\( \uparrow \gamma = 1. \)
\( \$ \) using Eq. (3.10) with \( a_{AA} = -3, a_{BB} = +2, a_{AB} = -2. \)

\[ \phi = \frac{M_{10}}{M_{10} + M_{01}} \quad (3.6) \]

\[ \langle x^2 \rangle = \frac{(1 - \phi)^2 M_{20} - 2\phi(1 - \phi)M_{11} + \phi^2 M_{02}}{M_{00}} \quad (3.7) \]

The content heterogeneity index can then be written as:

\[ \text{CHI} = \frac{\phi_B^2 M_{20} - 2\phi_A \phi_B M_{11} + \phi_A^2 M_{02}}{M_{00}} \left( \frac{M_{00}}{M_{10} + M_{01}} \right)^2 \quad (3.8) \]

where \( \phi_A, \phi_B \), are the overall mass fraction of solute and excipient, respectively (\( \phi_A + \phi_B = 1 \)). Therefore, a moments description of the bicomponent problem requires the tracking of the following mixed moments: \( M_{00}, M_{10}, M_{11}, M_{01}, M_{02} \) and \( M_{20} \).

### 3.5 Kernels

To study the effect of composition on the formation and size distribution of granules we introduce compositional dependence in the form:

\[ K_{12} = k(v_1, v_2)\kappa(c_1, c_2) \quad (3.9) \]
Here, $k(v_1, v_2)$ is the usual composition-independent kernel (a function of granule sizes $v_1$ and $v_2$) and $\kappa(c_1, c_2)$ is a function that depends on the composition (mass fraction of solute) of the two granules. This construction allows us to impose compositional effects on any composition-independent kernel. We adopt the following form for $\kappa$:

$$\ln \kappa = -a_{AA} c_1 c_2 - a_{BB} (1 - c_1)(1 - c_2) - a_{AB} (c_1 + c_2 - 2c_1 c_2)$$  (3.10)

The three parameters, $a_{AA}$, $a_{BB}$ and $a_{AB}$, control the strength of self aggregation (i.e., aggregation between pure solute or pure excipient) versus cross-aggregation (i.e., aggregation between pure solute and pure excipient). With $a_{AA} = a_{BB} = a_{AB} = 0$, Eq. (3.9) reverts to the classical kernel without any compositional effects. In the simulations shown here, $a_{AA} = -3$, $a_{BB} = +2$, and $a_{AB} = -2$. With this set of parameters, self aggregation of solute ($AA$ aggregation) is favored over self aggregation of excipient ($BB$) while the strength of cross aggregation of solute/excipient ($AB$) lies in-between. This provides qualitative analogy with an experimental system studied by Rajniak et al. [3]. In their study of the co-granulation of two excipients, mannitol and A-TAB, it was found that the two excipients tend to self aggregate, with mannitol exhibiting a higher rate relative to A-TAB. While the present study is not meant to simulate the system in Ref. [3], the selection of the parameters was guided by these data.

For the size dependent part of the kernel we have chosen two models:

Constant kernel \[ k(v_1, v_2) = K_0 \]

KTGF kernel \[ k(v_1, v_2) = K_0 \bar{v}^{\gamma} \left( \frac{1}{v_1} + \frac{1}{v_2} \right)^{1/2} \left( \frac{v_1^{1/3}}{1} + \frac{v_2^{1/3}}{2} \right)^2 \]

The constant kernel is a classic kernel that represents an idealized system in which all collisions are equally probable to result in aggregation. The so-called KTGF kernel is based on the kinetic theory of granular flow (KTGF) and is often used to model granulation [123]. The term $\bar{v}^{\gamma}$ accounts for the granular temperature of the system which increases with increasing size. While this factor affects the time evolution of the system, it has no effect on the distribution of components. The results shown here were obtained with either $\gamma = 0$ or $\gamma = 1$. In both kernels, $K_0$
is a constant numerical prefactor with units of m^3/s.

3.6 Methods of Solution

3.6.1 Rigorous Discrete Method

In this method the population balance is fully discretized and is represented by \( n_{ij} \), which gives the number concentration of particles that contain \( i \) units of solute and \( j \) units of excipient [15]. In the implementation shown here, we track 561 – 1920 classes which correspond to the maximum granule volume in the range 100 – 190.

The discrete version of the rigorous population balance equations for agglomeration in a one-component closed system [124] can be written as

\[
\frac{dn_i}{dt} = \frac{1}{2} \sum_{j=1}^{i-1} N_{i-j,j} - \sum_{j=1}^{N_C} N_{i,j}, \quad i = 1, 2, \cdots, N_C
\]

(3.13)

where \( N_{ij} \) is the rate of successful collisions between two particles with mass \( i \) and \( j \), respectively, and \( N_C \) is the number of size classes. For monovariate systems it is assumed that agglomerates (granules) are formed by lumps of primary particles that have approximately the same size. The size (volume) of any agglomerate is then characterized by the number of primary particles that form it and the size distribution is a discrete function, whose values \( n_i, i = 1, \cdots, N_C \) are the number concentrations of the agglomerates made by \( i \) primary particles. The maximum feasible size of a granule is determined by the number of classes \( N_C \), i.e., the number of primary particles in the largest granule. In this formulation, all masses are expressed as integer multiples of the unit mass. The number of classes (equations) \( N_C \) is in this case determined by the relative volumes of primary particles of excipient and solute and by the maximum feasible volume of the largest granule.

The discrete version of the bivariate population balance can be formally written
in a similar form as for the monovariate system:

\[
\frac{dn_{ij}}{dt} = \frac{1}{2} \sum_{k=1}^{i-1} \sum_{l=1}^{j-1} N_{i-k,j-l;k,l} - \sum_{k=1}^{\infty} \sum_{l=1}^{\infty} N_{i,j;k,l}
\]  

(3.14)

where \(N_{i,j;k,l}\) is the rate of agglomeration between particles that contain \(i\) units of solute and \(j\) units of excipient, and particles that contain \(k\) units of solute and \(l\) units of excipient:

\[
N_{i,j;k,l} = K(i,j;k,l)n_{ij}n_{kl}
\]  

(3.15)

More details and examples can be found in [125]. In the implementation shown here, we track \(N_C = 561\) to 1920 classes which correspond to the maximum granule volume in the range 100 to 190.

### 3.6.2 Constant-Number Monte Carlo (cNMC)

This methodology is based on event-driven Monte Carlo. Briefly, a pair of particles is selected for aggregation with probability \(p_{ij} = k_{ij}/k_{\text{max}}\), where \(k_{ij}\) is the aggregation kernel between the particles and \(k_{\text{max}}\) is the maximum kernel among the particles in the simulation. Upon aggregation, a new particle (granule) is formed that contains the solute and excipient of the parent particles. Before the procedure is repeated, a particle is selected at random and its copy is placed in the simulation box. This ensures that the number of particles in the simulation is constant at all time. Additional details of the algorithm have been given previously [106]. It was previously shown that good accuracy can be obtained using simulations with approximately 1000 particles [102]. Nevertheless, due to the complexity of the model being developed, the present simulations were conducted with 10000 particles.

### 3.6.3 DQMOM

In the direct quadrature method of moments (DQMOM) the population balance is converted into a set of ordinary differential equations for selected moments [90, 97, 126]. First, the distribution \(F(a, b)\) of granules containing \(a\) units of solute and \(b\) units of excipient is represented by a sum of \(N\) delta functions positioned at \((a_i, b_i)\)
with weights \( w_i, i = 1, \cdots, N \). The mixed moment of order \( k, l \) is expressed as

\[
M_{kl} = \sum_{n=1}^{N} w_n a_n^k b_n^l
\]

whose time derivative is

\[
\frac{dM_{kl}}{dt} = \sum_{n=1}^{N} \left[ (1 - k - l)a_n^k b_n^l \frac{dw_n}{dt} + k a_n^{k-1} b_n^l \frac{d(w_n a_n)}{dt} + l a_n^k b_n^{l-1} \frac{d(w_n b_n)}{dt} \right]
\]

This is equated to the evolution of the \( M_{kl} \) moment obtained from the population balance equation,

\[
\frac{dM_{kl}}{dt} = \frac{1}{2} \sum_{n=1}^{N} \sum_{q=1}^{N} w_n w_q \left[ (a_n + a_q)^k (b_n + b_q)^l - a_n^k b_n^l - a_q^k b_q^l \right] K_{n,q}
\]

where \( K_{n,q} \) is the aggregation kernel between masses \( a_n + b_n \) and \( a_q + b_q \). By applying the above equations to a base set of \( 3N \) moments, one obtains a closed system of ordinary differential equations for the unknown \( a_n, b_n, w_n (n = 1, \cdots, N) \). The calculations shown were performed using two different sets:

- **Set 1:** \( M_{00}, M_{10}, M_{20}, M_{30}, M_{40}, M_{50}, M_{01}, M_{21}, M_{41} \);
- **Set 2:** \( M_{00}, M_{10}, M_{01}, M_{20}, M_{11}, M_{02}, M_{30}, M_{21}, M_{12}, M_{03}, M_{31}, M_{13} \).

The first set was reported previously to lead to a numerically stable system [125]. This set, however, does not include certain moments that are required in the calculation of CHI from Eq. (3.8) and requires the estimation of the missing moments via approximate reconstruction of the distribution from its known moments. The second set is based on recent work by Fox [97]. It has the advantage that it is symmetric in both components and it contains all the moments required for the calculation of CHI.

In all cases concentrations are made dimensionless by their value at time zero. Time is made dimensionless based on the coagulation time \( \tau_C \), which is defined as

\[
\tau_C = \frac{1}{K_0 C_0},
\]
where \( C_0 \) is the initial number concentration of granules and \( K_0 \) is the numerical prefactor of the coagulation kernel.

### 3.7 Results

The cases investigated are summarized in Table 3.1 and they involve combinations of kernels with or without compositional effects. In most cases the initial distribution is bidisperse with a solute-to-excipient size ratio 1:10. In Case 6b, the size distribution of the excipient is normal with \( \bar{v}_B = 20 \) and variance \( \sigma^2 = 25 \); in case 6c it is lognormal with median 20 and geometric standard deviation 1.65 (the corresponding mean size is \( \bar{v}_B = 23 \)). In all of these cases, the number concentration of solute and excipient particles in the seed population are equal. Accordingly, the mass fraction \( \phi \) of solute varies somewhat depending on the details of the seed distribution. The values of \( \phi \) are listed in Table 3.1. In Case 1 and Case 2, the factor \( \kappa \) is set to 1, rendering the kernel in Eq. (3.9) independent of composition. All other cases incorporate that factor with the constants shown in Table 3.1.

The simplest case is that of the constant kernel with no compositional effects. The evolution of the average size as a function of time is shown in Fig. 3.1a. All methods are in excellent agreement up to about \( t = 100 \). Beyond this point, the discrete rigorous methods begin to fail because the maximum size is exceeded. Nonetheless, cNMC and DQMOM with either set of moments remain in excellent agreement. The content heterogeneity index is shown in Fig. 3.1b as a function of the mean size. Theory predicts [116] that CHI in this case is inversely proportional to the average size. The inverse relationship is captured very accurately by the rigorous methods as well as cNMC. When the set of 9 moments is used (Set 1), DQMOM fails to track the theoretical line. By contrast, the second set of 12 moments is in excellent agreement with the theoretical prediction. For the rest of the study, the first set of moments will be abandoned and all DQMOM calculations will implement Set 2.

In Case 2 the kernel is a pure KTGF kernel with no compositional effects (\( \kappa = 1 \)). The average size (Fig. 3.2a) is tracked very well by both cNMC and DQMOM. It is interesting to note that the long time evolution of the mean size has the same power-law scaling, \( \bar{v} \sim t^{6/5} \), as in one-component aggregation. Theory predicts [50]
that in this case the CHI scales as $1/\bar{v}$, a result that is confirmed in Fig. 3.2b. Here we also examine the effect of the maximum size on the accuracy of the rigorous method. The rigorous method is always accurate at early times; by increasing the maximum size, this accuracy extends to longer times. Fig. 3.2a shows this effect for an increase of the maximum size from 120 to 190. The improved accuracy comes at the expense of computational time. For the remainder of the paper, the calculation will be conducted with a maximum size in the range 110–190 unless
Figure 3.2: KTGF kernel – no compositional dependence (with $\gamma = 0$). Initial seeds bidisperse, $v_A = 1$, $v_B = 10$.

stated otherwise. This generally renders the rigorous method accurate up to $\bar{v} \approx 100$.

When compositional dependence is added to the constant kernel (Fig. 3.3) or to the KTGF kernel (Fig. 3.4), we continue to observe very good agreement between DQMOM and cNMC, both in $\bar{v}$ as well as in CHI. Unexpectedly, discrepancies begin to emerge when the KTGF kernel utilizes the granular temperature term
(γ = 1), as in Case 5 (see Fig. 3.5). This case differs from Case 4 only in the presence of the term \( \bar{v} \) in the kernel. This term has no effect whatsoever on the mixing of components and its sole contribution is to accelerate the overall rate of aggregation. Indeed, this is demonstrated by the fact that CHI, as calculated by cNMC or the rigorous method, is identical for both Case 4 and 5. Nonetheless, DQMOM fails to track the details for the mean size in the range \( t \approx 0.01 \) to 1, though it recovers at later times. These discrepancies increase when the details of the seed distribution are changed (Cases 6a–6c, and Figs. 3.6–3.8) and the long-time solution by the DQMOM often does not converge to that by cNMC.

3.8 Discussion

It is evident from the above comparisons that cNMC is a very accurate methodology for the simulation of bicomponent aggregation. It is in excellent agreement with the discrete population balance for all cases examined. It is also very robust as it does not require any special modifications for the various cases studied.

The performance of the DQMOM method is sensitive to a number of factors. Proper selection of the base set of moments is important. The original set of 9 moments provides excellent agreement for the mean granule size but fails for the content heterogeneity index, CHI. The most likely source for this discrepancy is the fact that the base set of 9 moments does not include \( M_{11} \) or \( M_{02} \), which are needed in CHI (Eq. (3.8)). The missing moments must be calculated by interpolation from the base set and this procedure clearly introduces error in the calculation of CHI. The expanded set of 12 moments includes those required in the calculation of \( \bar{v} \) and CHI; nonetheless, accuracy remains sensitive to the details of the kernel and the initial conditions. The KTGF kernel with compositional interactions causes DQMOM to lose accuracy with respect to the evolution of \( \bar{v} \). Increasing the initial size from \( v_A = 1 \) and \( v_B = 10 \) (Case 5) to \( v_A = 2 \) and \( v_B = 20 \) (Case 6a) causes DQMOM to over predict the mean size by a substantial factor and the errors persist as the details of the seed distribution are varied. Interestingly, the calculation of CHI is more accurate compared to \( \bar{v} \) and \( \langle x^2 \rangle \) less sensitive to the initial conditions. Apparently, errors in the computation of \( \bar{v} \) and \( \langle x^2 \rangle \) partially cancel out in the calculation of CHI, which is computed as \( \langle x^2 \rangle / \bar{v}^2 \).
Figure 3.3: Constant kernel with compositional dependence. Initial seeds bidisperse, $v_A = 1, v_B = 10$.

To summarize the results, cNMC is always found to be in excellent agreement with the rigorous computation of the discrete population balance. The method is straightforward to implement, fairly fast on a typical desktop unit, and requires no adjustments from one case to the next. The accuracy of the DQMOM method depends on the details of the problem examined. With proper selection of moments, the method is highly accurate if the kernel contains no compositional dependence. With explicit compositional effects, the accuracy of the method is variable depend-
Figure 3.4: KTGF kernel ($\gamma = 0$) with compositional dependence. Initial seeds bidisperse, $v_A = 1, v_B = 10$.

ing on the details of the seed distribution (Cases 3-6c). To understand the effect of compositional effects we consider the compositional distribution for Case 6b. Recall that in this case, the initial distribution consists of a monodisperse distribution of solute particles ($c = 1$) with size $v_A = 2$ and a normal distribution of excipient particles ($c = 0$) with mean size $v_B = 20$; the overall fraction of solute in the system is $\phi = 0.0909$. The growth rate during the initial stages of aggregation
Figure 3.5: KTGF kernel ($\gamma = 1$) with compositional dependence. Initial condition bidisperse with $v_A = 1$, $v_B = 10$.

is driven primarily by the presence of non-aggregated solute. These small solute particles have a higher rate of aggregation with larger particles as a result of the size dependence of the KTGF kernel but also due to the compositional factor $\kappa$, which promotes cross-aggregation. By $t \approx 1$ we find that all the solute particles have been consumed by aggregation and at this point the distribution consists of non-aggregated excipient, which is still present in significant amounts, and gran-
Figure 3.6: KTGF kernel ($\gamma = 1$) with compositional dependence. Initial condition bidisperse with $v_A = 2$, $v_B = 20$.

ules containing both components. The transition from a solute-rich regime to one in which the solute is depleted corresponds to the “bump” that is seen in the evolution of size between $t \approx 0.01$ and 0.1. This transition creates a dynamic that apparently causes DQMOM to lose accuracy.
Figure 3.7: KTGF kernel ($\gamma = 1$) with compositional dependence. Initial condition: solute particles are monodisperse with $v_A = 2$; excipient particles follow normal distribution with $\bar{v}_B = 20$ and variance $\sigma^2 = 25$.

3.9 Conclusions

With proper selection of the base set of moments, the DQMOM method is remarkably accurate when the kernel has no dependence on composition. When composition dependence is introduced, the accuracy of the method is variable and depends on the details of the initial distribution. By increasing the number of mo-
Figure 3.8: KTGF kernel \((\gamma = 1)\) with compositional dependence. Initial condition: solute particles are monodisperse with \(v_A = 2\); excipient particles follow lognormal distribution with \(\bar{v}_B = 23\), and geometric standard deviation \(s_g = 1.65\).

We are currently working in this direction. In the meanwhile, constant-number MC emerges as an accurate methodology in the study of bicomponent aggregation and in the validation of DQMOM.

4.1 Chapter Overview

The work presented within this chapter considers the impact of size, particle morphology, and binder composition to be key attributes that affect granulation [127]. We desire to add to the increasing body of granulation research that compares simulation results with experimental data. The originality of this work is that we seek to use one model to effectively simulate the growth profiles of three powders with varying degrees of surface roughness. A two-dimensional PB model that considers the impact of both solid powder and liquid binder is utilized; each granule is characterized by its total mass and the mass of binder it contains. We use constant-number Monte Carlo (cNMC) to solve the population balance equation. The discrete nature of cNMC makes it especially well-suited for multi-dimensional processes. Our research is complementary to the work of Oullion et al. [61] who use cNMC to compare simulation results with experimental data for high shear granulation with binder flow. However, this work seeks to evaluate the efficacy of a PB model that uses a physically based kernel to describe multiple fluidized bed granulation profiles; we also want to evaluate what the model reveals about
the coagulation constant and the binder distribution profile. Overall, we find that, through the use of a model with only one adjustable parameter, simulations match well with experimental data for three very different growth profiles.

4.2 Multi-Component Population Balance Model Development

The physical system that we model is a Wurster-style fluidized bed granulator [3]. Once a powder(s) is charged to the granulator, preheated air is metered to a stainless steel perforated plate at the bottom of the column. The Wurster unit has an internal central tube that facilitates a regular circulation pattern. Liquid binder is delivered via an upward-facing atomizing nozzle that is located at the bottom of the column within the inner tube. Because we treat both the powder and binder as explicit components in a granule, a multi-component mathematical model is needed to describe this system.

We focus on a two-component population balance model that describes the granulation of a single powder with binder. To model the granulation, we modify the Lushnikov equation [69] given in Chapter 2 to include a term that accounts for the continuous addition of binder droplets:

$$
\frac{\partial F(V_1, U_1)}{\partial t} = \frac{1}{2} \int_0^{V_1} \int_0^{U_1} K(V_1 - V_2, U_1 - U_2; V_2, U_2) \times
\left( F(V_1 - V_2, U_1 - U_2) F(V_2, U_2) dV_2 dU_2 - F(V_1, U_1) \int_0^{\infty} \int_0^{\infty} K(V_1, U_1; V_2, U_2) F(V_2, U_2) dV_2 dU_2 + \frac{\dot{B} \delta(V_1 - U_0, U_1 - U_0)}{m^3/s},
\right)
$$

(4.1)

where all terms are as defined before: $F(V_1, U_1)$ is the number density function (#/kg$^2$m$^3$) of particles with total mass $V_1$ (kg) and binder mass $U_1$ (kg), $F(V_1, U_1)dV_1dU_1$ represents the number concentration (#/m$^3$) of these particles and $K(V_1, U_1; V_2, U_2)$ represents the coagulation kernel (m$^3$/s) between two particles. During granulation, the total powder mass in the system is constant, but
binder mass increases due to the continuous flow of binder into the system. It is assumed that binder is introduced as mono-disperse droplets, however, other binder droplet distributions can be easily accommodated.

4.2.1 Kernel Description

The kernel used in this study is the KTGF kernel given in Chapter 2:

\[
K_{12} = K_0 k_{12}
\]  

(4.2)

with

\[
K_0 = \beta d_b^{\gamma+0.5}
\]  

(4.3)

\[
k_{12} = \psi_{12} x_{13}^{-\gamma} z_{12}
\]  

(4.4)

\[
z_{12} = \left(\frac{v_1^{1/3} + v_2^{1/3}}{v_2} \right)^2 \left(\frac{1}{v_1} + \frac{1}{v_2} \right)^{1/2}.
\]  

(4.5)

The \(K_0\) term is a coagulation rate constant that contains all dimensional parameters and has units of m^3/s; \(k_{12}\) contains the size-dependent portion of the kernel.

We carefully examined the nature of \(K_0\) for the experimental system under consideration. In a Wurster fluidized bed reactor, the binder solution is forcefully directed into the fluidized bed via an atomizing nozzle. As such, the potential need for three \(K_0\) values (one each for binder droplet-binder droplet coagulation, binder droplet-particle coagulation, and particle-particle coagulation) was considered. However, due to the spray nozzle position, its orientation and the relative rate of binder spray into the granulator compared with the particle velocity in the bed, we assumed that binder droplet-binder droplet interactions were negligible and binder-particle interactions occurred much faster than did particle-particle interactions. We thus adopted the following convention for setting the value of \(K_0\) for each coagulation event:

\[
K_0 = 0 \text{ for binder-binder coagulation}
\]

\[
K_0 \text{ for binder-particle coagulation} = 100K_0 \text{ of particle-particle coagulation}
\]
In essence, binder-binder droplets were prohibited and the $K_0$ for binder-particle interactions was set arbitrarily high such that, whenever binder particles were present in the simulation box, binder-particle interactions would be more probable. Thus, only one value for $K_0$ is required.

### 4.2.2 Composition-Dependent Portion of Kernel

The success factor, $\psi_{12}$, used in this study is the same as that given in Chapter 2:

$$\psi_{12} = \psi_{\text{geom}} \psi_{\text{phys}}.$$  \hspace{1cm} (4.6)

To recap, the geometrical portion considers the probability that two granules will hit each other on a wet surface based on microscopic models for binder-granule dynamics [2, 15, 83]. The physical portion of the success factor considers whether or not the inertial forces associated with a collision can be dissipated by the viscous forces associated with the binder properties and the thickness of the binder on the particle surfaces. If the inertial forces can be dissipated, then the two colliding particles remain stuck together, the particles rebound otherwise [6, 15, 57, 82].

### 4.3 Development of Constant-Number Monte Carlo with a Flow Stream

The numerical solution of the population balance equation is accomplished via constant-number Monte Carlo [99, 100]. This methodology is easily extended to any number of components and is especially simple to implement in a batch system. Open systems can be incorporated, as has been demonstrated previously [101, 107]. The simultaneous incorporation of flow streams and multiple components (excipient, binder) is explained in more detail below.

In Monte Carlo, each process that affects the particle distribution is simulated as a stochastic event. For the purposes of this analysis, “particle” refers to binder drops, dry granules (primary particles) and binder-covered granules of any size.
These are distinguished internally in the simulation by their composition, i.e., the percentage of mass due to binder and solid excipient. There are two such processes in this problem: agglomeration, which refers to the formation of a larger granule from two smaller ones, and binder flow, which adds binder "particles" into the granulator. As before, the basic outline of the algorithm is as follows:

1. Calculate the probability of each event (e.g., introduction of binder or agglomeration).

2. Choose the event to occur via a uniform random number generator.

3. Implement the selected event as follows. If the event is agglomeration, two particles are selected and combined into a new granule. A randomly selected particle is then duplicated and added into the simulation box to keep the number constant. If the event is binder flow, a predetermined number of binder drops are introduced. An equal number of particles, chosen randomly, are deleted from the simulation box to keep the total number constant.

4. Update all metrics (mass concentration, number concentration, time advancement, etc.)

These items are described in more detail in the following sections.

4.3.1 Event Rates and Probabilities

The probability of an event is calculated in proportion to the rate of the event by the equations

\[ P_{coag} = \frac{r_{coag}}{r_{coag} + r_{FS}} \]  \hspace{1cm} (4.7)

\[ P_{FS} = \frac{r_{FS}}{r_{coag} + r_{FS}} \]  \hspace{1cm} (4.8)

where \( P_{event} \) is the probability and \( r_{event} \) the rate of a particular Monte Carlo event (the subscript “FS” stands for flow stream). The dimensionless coagulation rate is as defined in Chapter 2:

\[ r_{coag} = \frac{(k_{12}) c_p^2}{2} \]  \hspace{1cm} (4.9)
where $\langle k_{12} \rangle$ is the mean dimensionless coagulation kernel between all particle pairs and $c_p$ is the dimensionless number concentration of particles defined as $c_p = C_p/C_0$.

The flow of binder is discretized by defining the “flow event” as the introduction of $n$ binder drops into the granulator ($n$ is specified in Table 4.2). The dimensionless rate of this event is

$$r_{FS} = \frac{t^*}{n\tau_{FS}}$$

(4.10)

where $t^*$ is already given as $\tau_{coag}$ in section 2.5.4.3 and $\tau_{FS}$ is the time constant for flow (s), defined as

$$\tau_{FS} = \frac{V_R}{V_*\dot{Q}}$$

(4.11)

where $V_R$ is the reactor volume ($m^3$), $V_*$ is a characteristic volume given as $1/C_0$ in section 2.5.4.3, and $\dot{Q}$ is the flow rate (1/s) of binder droplets into the reactor. The parameter $n$ may be viewed as a coarsening factor. Maximum accuracy for the given number of simulation particles is obtained by setting $n = 1$, however, this may result in longer execution times. We treat $n$ as an adjustable parameter whose optimum value is determined by trial and error.

### 4.3.2 Event Implementation

Events are implemented by the general constant-number algorithm [99–101]. For coagulation, a pair of granules is elected with probability proportional to the dimensionless kernel $k_{ij}$ between them. Upon coagulation, a new granule is formed whose constituent components (solid excipient and binder) is the sum of those in the parent granules. The constant number of simulation particles is enforced by duplicating a granule at random and placing the copy in the simulation box. If the chosen event is addition of binder via the binder flow stream, then $n$ randomly selected particles are chosen via a uniform random number generator and removed from the simulation box. In their place, $n$ binder particles are introduced.
4.3.3 Concentrations

The number and mass concentration are calculated as in Chapter 2. The dimensionless mass concentration, $c_m$, is:

$$c_m = c_{m,\text{old}} \left(1 + \frac{\delta v_{\text{event}}}{N\bar{v}}\right).$$

(4.12)

Here, $c_{m,\text{old}}$ is the dimensionless mass concentration from the previous Monte Carlo step, $N$ is the number of simulation particles, $\bar{v}$ is the mean dimensionless particle mass in the simulation box, and $\delta v_{\text{event}}$ is the change in the mass of the simulation box as a result of the event chosen. For coagulation, $\delta v_{\text{event}} = 0$ (no net change of mass); for a binder event, $\delta v_{\text{event}} = n v_b$, where $n$ is the number of binder particles introduced, and $v_b$ is the dimensionless mass of a binder particle. The dimensionless number concentration is calculated as

$$c_p = \frac{c_m}{\bar{v}},$$

(4.13)

where $c_m$ is the mass concentration calculated above and $\bar{v}$ is as defined previously.

4.3.4 Time Increment

The mean dimensionless time that elapses between Monte Carlo events is given as in Chapter 2:

$$\Delta \tau = \frac{c_p}{N r_{\text{coag}} + r_{FS}}$$

(4.14)

and is calculated based on the current values of the number concentration, $c_p$, and event rates, $r_{\text{coag}}$, $r_{FS}$. A summary of the important equations necessary to conduct constant-number Monte Carlo for the current scenario is given in Table 4.1.

4.4 Validation of cNMC algorithm

All simulations were conducted using FORTRAN and a computer with a 2.80 GHz processor and 1.00 GB of RAM. Before comparing simulation results with experiment, we first validated the model by examining the case of continuous addition of binder with simplified kernels. We then compared the cNMC results
Table 4.1: Key constant-number Monte Carlo equations

<table>
<thead>
<tr>
<th>Quantity</th>
<th>Equation</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Coagulation</strong></td>
<td></td>
</tr>
<tr>
<td>Rate</td>
<td>(r_{\text{coag}} = \frac{\langle k_{12} \rangle c_p^2}{2})</td>
</tr>
<tr>
<td>(\tau_{\text{event}})</td>
<td>(\tau_{\text{coag}} = \frac{1}{K_0C_0})</td>
</tr>
<tr>
<td>Probability</td>
<td>(P_{\text{coag}} = \frac{r_{\text{coag}}}{r_{\text{coag}} + r_{\text{FS}}})</td>
</tr>
<tr>
<td>(\delta v_{\text{event}})</td>
<td>0</td>
</tr>
<tr>
<td><strong>Binder Flow Stream</strong></td>
<td></td>
</tr>
<tr>
<td>Rate</td>
<td>(r_{\text{FS}} = \frac{\tau_{\text{coag}}}{\tau_{\text{FS}}} n)</td>
</tr>
<tr>
<td>(\tau_{\text{event}})</td>
<td>(\tau_{\text{FS}} = \frac{V_RC_0}{Q})</td>
</tr>
<tr>
<td>Probability</td>
<td>(P_{\text{FS}} = \frac{r_{\text{FS}}}{r_{\text{coag}} + r_{\text{FS}}})</td>
</tr>
<tr>
<td>(\delta v_{\text{event}})</td>
<td>(nv_b)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Quantity</th>
<th>Equation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mass concentration</td>
<td>(c_m = c_{m,\text{old}} \left(1 + \frac{\delta v_{\text{event}}}{N \bar{v}}\right))</td>
</tr>
<tr>
<td>Number concentration</td>
<td>(c_p = \frac{c_m}{\bar{v}})</td>
</tr>
<tr>
<td>Time between events</td>
<td>(\Delta \tau = \frac{c_p}{N} \frac{1}{r_{\text{coag}} + r_{\text{FS}}})</td>
</tr>
</tbody>
</table>

with an analytical solution, if available, or solutions from other numerical solution techniques.

### 4.4.1 Case 1: Constant Kernel

If the aggregation kernel is constant, the number and mass concentration of particles (including granules and binder drops) is given by the simplified equations

\[
\frac{dc_p}{d\tau} = \frac{\tau_{\text{coag}}}{\tau_{\text{FS}}} - \frac{c_p^2}{2} \tag{4.15}
\]

and

\[
\frac{dc_m}{d\tau} = \frac{\tau_{\text{coag}}}{\tau_{\text{FS}}}. \tag{4.16}
\]
Figure 4.1: Average granule size versus time for constant kernel. Average size was made dimensionless by monodisperse binder droplet size. Binder-binder collisions were permitted.

An analytical solution for the average granule size as a function of time is determined from the solution of these equations. Figure 4.1 shows the results for $\frac{\tau_{\text{coag}}}{\tau_{\text{FS}}} = 0.0608$. As we see, the agreement between cNMC and the analytical solution is very good. These profiles are also compared with the rigorous discrete method (labeled as RM). Briefly, in the rigorous discrete method the population balance is fully discretized and is represented by $n_{ij}$, which gives the number concentration of particles that contain $i$ units of powder and $j$ units of binder [15]. The maximum feasible size of a granule is determined by the number of classes used in the simulation. (The number of classes is also identical with the number of solved ordinary differential equations). Increasing the number of classes in the simulation extends the accuracy of this numerical technique to longer times, but at the expense of simulation speed. A more thorough discussion of this numerical solution technique may be found in our previous work [128]. The main point to note here is that, as the number of classes is increased, the RM solution converges to the analytical solution.
Figure 4.2: Average granule size versus time for KTGF kernel with geometric success factor. Average size was made dimensionless by monodisperse binder droplet size. Binder-binder collisions were prohibited. Particle-binder $K_0$ is 100 times greater than particle-particle $K_0$.

### 4.4.2 Case 2: KTGF Kernel

We conducted a final validation check by evaluating the case of a simplified KTGF kernel with only a geometric success factor. (The main difference between this case and the model we use to compare with experiments is the absence of a physical success factor here). Because no analytical solution exists for this case, we proceeded by comparing the granulation profiles of the rigorous discrete method and cNMC. As before, we consider the rigorous method to be accurate at early times; by increasing the maximum size, this accuracy extends to longer times. In Fig. 4.2, we note that the rigorous discrete method converges to the cNMC results with increasing number of size classes. However, at this level, the rigorous method is already very computationally expensive; simulation times are on the order of several hours whereas cNMC simulation times are on the order of several minutes. Nevertheless, the observed convergence lends yet another degree of credibility to the development of the cNMC model.
4.5 Results and Discussion

The experiments that form the basis for assessing the current granulation model are those of Rajniak et al. [3, 15], which examined the granulation profile of three excipients: microcrystalline cellulose (Avicel Ph102), mannitol (Pearlitol SD200) and dicalcium phosphate (A-TAB). The experiments are briefly reviewed below. Narrow sieve fractions (0.920 kg) of an excipient were granulated with a 15 wt.% aqueous solution of hydroxypropyl cellulose binder (HPC) in a bottom-spray Wurster type fluidized bed granulator. The binder spray rate was approximately 11-13 g/min, the fluidizing air flow rate (1.4-2.3 cubic meters per minute) was gradually increased throughout the experiment to maintain a good and consistent flow pattern, and the average temperature of the fluidized bed was maintained at 19-22°C.

The parameters used in the simulations are summarized in Table 4.2. The parameter $\gamma$ is not included because, when correlating simulation results to experimental data, it was determined that the best agreement in all cases occurred when $\gamma$ was set equal to zero (results not shown). Non-zero values caused an increased convexity in the granulation growth profile that deviated from the experimental data points. Since the experimental data could be fully explained without this term, $K_0$ became the only adjustable parameter.

Experimental $d_{43}$ values are compared with $d_{43}$ values obtained via Monte Carlo simulation in Figures 4.3a-4.3c for A-TAB, Mannitol, and Avicel, respectively. The simulation $d_{43}$ is calculated based on powder mass and dried binder (i.e., dried granules with all moisture removed). The optimal simulation $K_0$ values that best matched the experiments are listed in Table 4.3.

In the case of A-TAB, a relatively long lag time is present before the granules start to grow. This delay in the onset of growth is attributed to the rough surface of A-TAB particles; sufficient binder must first accumulate in the crevices of the particle before granule growth can commence. The simulation results capture this feature quite well.

The growth of the powder with an intermediate degree of surface roughness, Mannitol, is shown in Fig. 4.3b. This excipient shows a slow, but steady growth rate initially. Approximately 15 minutes into the granulation, the growth rate increases to a new steady state value. Simulation results indicate that a moderate
Table 4.2: Simulation parameters

<table>
<thead>
<tr>
<th>Variable</th>
<th>Value</th>
<th>Units</th>
<th>Reference/Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>N</td>
<td>10000</td>
<td>—</td>
<td>Arbitrary selection</td>
</tr>
<tr>
<td>n</td>
<td>10</td>
<td>—</td>
<td>Optimized selection</td>
</tr>
<tr>
<td>$d_b$</td>
<td>60</td>
<td>µm</td>
<td>[3]</td>
</tr>
<tr>
<td>$\rho$</td>
<td>1021</td>
<td>kg/m$^3$</td>
<td>Given by experimentalist</td>
</tr>
<tr>
<td>$\mu$</td>
<td>0.762</td>
<td>kg/(m s)</td>
<td>[3]</td>
</tr>
<tr>
<td>$u_0$</td>
<td>5.5</td>
<td>m/s</td>
<td>Given by experimentalist</td>
</tr>
<tr>
<td>$V_R$</td>
<td>0.01</td>
<td>m$^3$</td>
<td>10.0 L granulator</td>
</tr>
<tr>
<td>$r_g$</td>
<td>60</td>
<td>µm</td>
<td>[2]</td>
</tr>
<tr>
<td>Mannitol $C_0$</td>
<td>$7.91 \cdot 10^{10}$</td>
<td>1/m$^3$</td>
<td>Calculated value</td>
</tr>
<tr>
<td>A-TAB $C_0$</td>
<td>$3.07 \cdot 10^{10}$</td>
<td>1/m$^3$</td>
<td>Calculated value</td>
</tr>
<tr>
<td>Avicel $C_0$</td>
<td>$3.30 \cdot 10^{10}$</td>
<td>1/m$^3$</td>
<td>Calculated value</td>
</tr>
<tr>
<td>Initial Mannitol $d_{43}$</td>
<td>129.6</td>
<td>µm</td>
<td>[15]</td>
</tr>
<tr>
<td>Initial A-TAB $d_{43}$</td>
<td>177.5</td>
<td>µm</td>
<td>[15]</td>
</tr>
<tr>
<td>Initial Avicel $d_{43}$</td>
<td>173.4</td>
<td>µm</td>
<td>[15]</td>
</tr>
<tr>
<td>Mannitol and A-TAB $\dot{Q}$</td>
<td>$1.693 \cdot 10^6$</td>
<td>1/s</td>
<td>Calculated value</td>
</tr>
<tr>
<td>Avicel $\dot{Q}$</td>
<td>$1.485 \cdot 10^6$</td>
<td>1/s</td>
<td>Calculated value</td>
</tr>
</tbody>
</table>

A coating period exists until about 10 minutes into the granulation; thus, the simulation slightly underpredicts the experimental values at this stage. After this point, however, the simulation results show good agreement with the experimental data through the end of the granulation.

Figure 4.3c shows the growth profile of Avicel, the powder with the smoothest surface of the three. Experimental values show a fast growth rate that begins immediately and remains high throughout the granulation. Once again, Monte Carlo simulations indicate a coating period, albeit a marginal one. At this early stage until approximately halfway through the granulation, simulation results slightly underpredict granule size. At that time, the simulation slightly overpredicts the $d_{43}$ given by experiment. Nevertheless, the simulation still compares favorably with the experimental values in both overall growth rate and granule size.
4.5.1 $K_0$ Analysis

Though the simulation $K_0$ values are within one order of magnitude of each other, the differences are significant enough to suggest that some element important to the granulation process is still unaccounted for in the kernel. We hypothesize that hygroscopicity may explain the differences. More hygroscopic powders draw relatively more water from the binder than would less hygroscopic powders, thus causing the binder to dry faster. As the binder dries, a point is reached where the binder is no longer effective (i.e., it no longer promotes successful collisions). If this effect is not considered within the kernel, then the apparent $K_0$ would be lower for the more hygroscopic powders. As noted in Table 4.3, the order of decreasing $K_0$
Table 4.3: Optimized values of $K_0$

<table>
<thead>
<tr>
<th>Powder</th>
<th>$K_0$ Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mannitol</td>
<td>$2.5 \cdot 10^{-14}$ m$^3$/s</td>
</tr>
<tr>
<td>Avicel</td>
<td>$7.0 \cdot 10^{-14}$ m$^3$/s</td>
</tr>
<tr>
<td>A-TAB</td>
<td>$12.5 \cdot 10^{-14}$ m$^3$/s</td>
</tr>
</tbody>
</table>

is A-TAB > Avicel > Mannitol. Based on the above hypothesis, we would expect Mannitol to be the most hygroscopic of the three powders. Sorption isotherms conducted by Merck & Co., Inc. indicate that the degree of hygroscopicity for the three excipients is indeed Mannitol > Avicel > A-TAB. To further test this hypothesis, we need a kernel that incorporates the effect of moisture into the success factor. The current kernel does not include this component.

### 4.5.2 Binder Distribution Analysis

A benefit of mathematical modeling in general, and cNMC in particular, is that we can obtain more information about a system than is obvious or easily gathered from experiment. One such example is the distribution of binder among the various size classes during granulation. In Fig. 4.4 is shown the binder distribution for all three powders at various times of the simulation. The left column in the figure corresponds to Avicel granulation, the middle column corresponds to Mannitol and the right column corresponds to A-TAB. Each plot shows binder mass fraction as a function of granule size. The bar at the top of each figure, in the bin containing particles in the lowest size class and a binder mass fraction of 1, corresponds to the presence of unused binder droplets in the simulation box.

Granule growth occurs rapidly for the smoothest powder, Avicel. Even at a relatively early time of $t = 15$ minutes, significant granule growth has already occurred (see Fig. 4.4a). It is instructive to note that the large granules grew quickly despite having a low binder mass fraction. By the end of the simulation (which also corresponds to the end of binder addition), all size classes contain more binder than at earlier points in the simulation; the smallest particles contain the
Figure 4.4: Binder distribution as a function of time. The binder mass fraction axis is linear and ranges from 0 to 1, whereas the size axis is logarithmic and ranges from a dimensionless size of 0 to approximately 20,000 \( (d_{43} \approx 1,630 \mu m) \). For each size class, the height of each bar is only relative to the other bars within that size class.

The opposite is true for the roughest powder, A-TAB; these powder particles require significant binder coating before coagulation can occur. Fifteen minutes into the granulation, granule growth is still virtually nonexistent (<0.5% of the powder has combined, see Fig. 4.4c). Since granule growth generally occurs via particles that have binder with a threshold binder mass fraction of approximately 0.3, the newly formed granules retain this attribute. It is for this reason that, once granule growth commences, the larger granules remain close to this critical threshold (typically within a range of 0.10, see Fig. 4.4f). As in the case of Avicel,
at the end of the simulation the smaller granules have a larger binder mass fraction due to the lengthy exposure of these particles to the binder spray (see Fig. 4.4i).

Because the primary particle roughness of Mannitol lies between that of Avicel and A-TAB, the binder distribution profile also lies between that observed for Avicel and Mannitol. Overall, these results generally complement the work of others [59,65,129] that show a non-uniformity of binder distribution for both wet high shear granulation and fluidized bed granulation.

### 4.5.3 Coarsening Factor Analysis

In section 4.3.1, we defined a binder flow event as the addition of \( n \) droplets of binder into the granulator. This value appears to be arbitrary. However, the proper use of coarsening is to strike a balance such that simulation speed is optimized without sacrificing accuracy. To provide guidance on how to judiciously select an appropriate value for the coarsening factor, we studied its impact on the simulations of all three powders studied. While we only present the results associated with Mannitol, the same general observations and conclusions could be made for all three powders.

The impact of the coarsening factor on simulation speed is shown in Fig. 4.5a. The approximate time to conduct a simulation drops from 1 to 10 binder drops per event and then tends to plateau before the relationship deteriorates at high coarsening factors. The gain in speed is primarily explained by a reduction in the number of calculations demanded by the computer code. At some point, however, the time benefit from a decreased number of binder flow events becomes offset by the sheer quantity of the coagulation events.

In Fig. 4.5b, we examine the impact of the coarsening factor on simulation accuracy. The granulation profile for coarsening factors of 1 and 10 are nearly indistinguishable. At coarsening factors at or above 1,000, the granulation profile becomes more distinct. This behavior can partially be explained by examining the event rates and their relative values. Because binder must be present on the powder surface before granule growth can occur, the first event in all cNMC simulations must always be a binder flow event. A coarsening factor of 9,000 binder droplets
corresponds to a time step for this first event of approximately 40 minutes. Thus, after only one Monte Carlo event, and before any coagulation is permitted to occur, we have already jumped forward several minutes into the simulation. Such a large time step obviously misses many coagulation events that could have occurred along the way.

The coarsening factor of choice for these simulations was determined to be 10 binder droplets per event. This value minimized the computation time while still maintaining the accuracy obtained by the use of smaller coarsening factors. The main governing factor in choosing this value is the relative rate of the competing events (i.e., coagulation and binder flow events). The ultimate goal is to have rates that are close in value such that the “delta $t$” between each Monte Carlo event is comparable. Ideally, these values should be within an order of magnitude of each other.

4.6 Conclusions

We adapted a multidimensional population balance model to include the effect of particle morphology on granulation growth. We accomplished this task through the use of a semi-batch constant-number Monte Carlo method and the implementation of a physically based description for the impact of binder content on the coagulation kernel’s collision success factor. cNMC simulations were fitted to experimental data; the binder distribution associated with each growth profile was explained by the degree of primary particle surface roughness. The current model has great potential despite the fact that it currently does not consider the impact of moisture on the granulation profile. Through the use of only one adjustable parameter, the agglomeration constant of the KTGF kernel for particle-particle collisions, simulations match well with experimental data for three very different growth profiles.
Figure 4.5: Analysis of coarsening factor (number of binder droplets introduced per event) impact on simulation speed and accuracy for Mannitol. A typical simulation run time for several different coarsening factors is given in 4.5a. Average granulation growth profiles for several coarsening factors are shown in 4.5b (average of 10 simulation runs).
Chapter 5

Continuous Addition of Binder with Moisture Removal

5.1 Chapter Overview

In this chapter, the impact of moisture on granulation is considered. In particular, the binder is now treated explicitly as a bicomponent solution of soluble binder in water. Accordingly, a granule consists of three components, solid powder, soluble binder and water, and the population balance becomes a three-component problem. The granulator is operating under semi batch conditions, with the solid material preloaded and the binder sprayed continuously while air is drawn out to maintain the pressure in the granulator. Moisture enters and exits the granulator via the air stream. In addition, a certain amount of moisture evaporates from the wet droplets. Thus, in addition to the usual PBE, a moisture balance must now be included. This chapter details the governing equations surrounding this process and the modifications that must be made to the traditional constant-number Monte Carlo technique so that moisture content can be tracked both within granules and in the ambient air within the granulator. Solution of the population balance equation via Monte Carlo is then validated by comparison with the rigorous discrete method described in Chapter 2. This chapter continues with a discussion of the temperature impact on the granule growth rate and moisture content before concluding with a comparison of simulation results with experimental values and an
examination of the nuances associated with the Monte Carlo model. It is found
that Monte Carlo can effectively simulate the impact of moisture on granulation.

5.2 Theoretical Background

During fluid bed granulation, the air used to fluidize the bed also serves to dry
the granules. The rate of binder drying directly affects the granulation growth
profile. As moisture is removed from a granule via evaporation, the binder will
eventually reach a point where it is no longer effective in promoting granulation
(i.e., the binder is no longer “sticky” and colliding granules rebound rather than
stick together) unless fresh binder is added to the granule. Research has been
conducted on the effect of processing variables such as bed temperature and flu-
idization air velocity [130,131] on granule growth rates within a fluidized bed, but
this work was often experimental in nature. It has not been until recently that
mechanisms associated with fluidized bed granulation such as binder solidification
were incorporated into models [35,132]. The research presented within this chapter
seeks to continue this development by coupling the impact of binder solidification
via evaporation with a complex, physically based coagulation kernel that may be
used to model granulation for the kind of particles typically encountered in the
pharmaceutical industry (i.e., particles with a wide range of surface asperities).

The physical system under examination is a fluidized bed granulator with an
inlet feed of binder droplets and warm air with some amount of moisture, an out-
let stream of cooler air with more moisture relative to the inlet, and in which
granulation and evaporation occur. A schematic of this system is shown in Figure
5.1. Any model that considers these four simultaneous processes must adequately
describe the coagulation kernel and the evaporation rate. Because a Wurster flu-
idized bed granulator is the system under study, the size-dependent portion of the
KTGF kernel described in Chapter 2 is still utilized. However, the success factor
is modified to account for any drying of the binder that occurs. The overarching
equations that govern these processes are described in the sections below.
5.2.1 Governing Equations

A mathematical description of the fluidized bed granulator requires the tracking of three quantities: the number of particles, the mass of the particles, and the moisture content of the air within the granulator. As such, balances are needed for the following dimensional concentrations:

- \( C_p \) number concentration of particles (m\(^{-3}\))
- \( C_m \) mass concentration of particles (kg/m\(^3\))
- \( C_w \) mass concentration of moisture in air (kg/m\(^3\))

“Particles” refer to either binder drops, dry granules, or wet granules. As far as the simulation is concerned, the only difference between these species is their composition. The composition of particles is tracked by the simulation, therefore, no
equations need to be written for the distribution of these multicomponent particles. The granulator is modeled as a well-mixed reactor of volume $V_R$ ($m^3$). The balance equations for particles and moisture are:

$$\frac{dC_p}{dt} = -\langle K_{12} \rangle \frac{C_p^2}{2} + \frac{\dot{Q}}{V_R} \quad (5.1)$$

$$\frac{dC_m}{dt} = \frac{\dot{Q} M_p0}{V_R} - \dot{M}_{evap} \quad (5.2)$$

$$\frac{dC_w}{dt} = \frac{Q_{in} C_{w0}}{V_R} - \frac{Q_{out} C_w}{V_R} + \dot{M}_{evap} \quad (5.3)$$

for the number concentration of particles ($#/m^3 s$), mass concentration of particles ($kg/m^3 s$) and mass concentration of moisture in the air ($kg/m^3 s$), respectively. The various parameters are defined below:

$\langle K_{12} \rangle = K_0 \langle k_{12} \rangle$ mean agglomeration kernel among all particles ($m^3/s$)

$K_0$ dimensional kernel (constant) ($m^3/s$)

$\langle k_{12} \rangle$ dimensionless kernel (function of particle properties) (-)

$Q_{in}$ volumetric flow of air in ($m^3/s$)

$Q_{out}$ volumetric flow of air out ($m^3/s$)

$\dot{Q}$ spray rate of binder drops (1/s)

$C_{n0}$ $\equiv \dot{Q}/Q_{in}$ ($1/m^3$)

($#$ of binder droplets coming in per $m^3$ of air in)

$M_{p0}$ mass of monodisperse binder droplet in inlet (kg)

$C_{w0}$ moisture concentration in inlet (kg/m$^3$)

$C_w$ moisture concentration in reactor (kg/m$^3$)

$\dot{M}_{evap}$ total rate of evaporation (kg/m$^3 s$)

By defining $C_{n0} = \dot{Q}/Q_{in}$, both inlet streams (air, binder spray) are described by a single time constant. The fact that these streams are physically separate is of no relevance to the simulation. It is also pertinent to note that both the inlet and outlet volumetric flows may be different if so desired (although in many practical cases of interest these flow rates are equal).
Table 5.1: Dimensionless variables

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$c_p = C_p/C_0$</td>
<td>number concentration of particles in reactor</td>
</tr>
<tr>
<td>$c_{n0} = C_{n0}/C_0$</td>
<td>number concentration of (binder) particles in inlet</td>
</tr>
<tr>
<td>$c_m = C_m/C_0M_b$</td>
<td>mass concentration of particles</td>
</tr>
<tr>
<td>$c_w = C_w/C_0M_b$</td>
<td>mass concentration of moisture in reactor</td>
</tr>
<tr>
<td>$c_{w0} = C_{w0}/C_0M_b$</td>
<td>mass concentration of moisture in inlet</td>
</tr>
<tr>
<td>$v_b = M_{p0}/M_b$</td>
<td>mass of single binder drop in inlet</td>
</tr>
<tr>
<td>$\dot{m}<em>{\text{evap}} = M</em>{\text{evap}}\tau_C/C_0M_b$</td>
<td>evaporation rate</td>
</tr>
<tr>
<td>$\tau = t/\tau_C$</td>
<td>time</td>
</tr>
<tr>
<td>$\tau_{\text{in}} = V_R/Q_{\text{in}}\tau_C$</td>
<td>inlet flow time</td>
</tr>
<tr>
<td>$\tau_{\text{out}} = V_R/Q_{\text{out}}\tau_C$</td>
<td>outlet flow time</td>
</tr>
</tbody>
</table>

To write the balances in dimensionless form, the following characteristic quantities are used: number concentration $C_*$ (1/m³), mass $M_*$ (kg) and time $t_*$ (s). The values for characteristic number concentration and time given in Chapter 2, namely $C_0$ and $\tau_{\text{coag}}$, respectively, are used. A characteristic mass equal to the mass of a monodisperse binder droplet, $M_b$ is now included in the list of characteristic quantities. Using these constants, the balance equations in dimensionless form are:

$$\frac{dc_p}{d\tau} = -\left\langle k_{12} \right\rangle \frac{c_p^2}{2} + \frac{c_{n0}}{\tau_{\text{in}}} \quad (5.4)$$

$$\frac{dc_m}{d\tau} = \frac{c_{n0}v_b}{\tau_{\text{in}}} - \dot{m}_{\text{evap}} \quad (5.5)$$

$$\frac{dc_w}{d\tau} = \frac{c_{w0}}{\tau_{\text{in}}} - \frac{c_w}{\tau_{\text{out}}} + \dot{m}_{\text{evap}} \quad (5.6)$$

A detailed discussion on the derivation of these dimensionless equations may be found in section 5.4. All dimensionless variables in these equations are defined in Table 5.1.

The characteristic coagulation time with units of seconds is given as $\tau_C = 1/K_0C_0$. Note that the characteristic inlet flow time, $\tau_{\text{in}}$ differs from that given in Chapter 4. This change is intentional and was made to align the characteristic inlet flow time with the newly required characteristic outlet flow time.
5.2.2 Incorporation of Binder Drying Effect

Once a binder droplet is introduced to the system, drying can immediately begin. This drying serves to continuously increase that droplet’s viscosity. If the binder is initially very dilute, then the binder may be too watery to dissipate the inertial force associated with collision. However, as the binder dries, its viscosity increases. At some point, a critical viscosity is reached where granule growth can commence. In the experiments considered within this dissertation, the introduced binder droplets are already at this point. As drying proceeds, the binder continues to become more viscous and eventually reaches a maximum effectiveness where coagulation is no longer permitted without the introduction of fresh binder. In essence, the granule reverts back to a dry granule.

Mathematically, this drying process is taken into account in the physical portion of the success factor described in Chapter 2. Briefly, the physical success factor evaluates, via a Stokes number analysis, if enough viscous force is present to prevent particle rebound due to inertial forces. Low viscosity corresponds to a large Stokes number and vice versa. When the Stokes number is less than a critical value, coagulation is permitted to occur. Increasing viscosity always leads (mathematically) to a smaller Stokes number and greater probability for a successful collision. There is no provision to prevent coagulation between granules with binder that is too dry. As such, a modification to the success factor is proposed that considers the internal distribution of moisture within a granule.

The success factor $\psi_{12}$ is incorporated in the coagulation kernel. To recap, the kernel used is:

$$K_{12} = K_0 k_{12}$$

where $K_0$ is a dimensional rate constant with units of $m^3/s$ and $k_{12}$ contains the composition ($\psi_{12}$) and size dependence ($z_{12}$) of the KTGF kernel:

$$k_{12} = \psi_{12} z_{12}$$

$$z_{12} = \left(\frac{v_1^{1/3} + v_2^{1/3}}{v_1 + v_2}\right)^{1/2}.$$  

To include the impact of changing viscosity on the success factor, a third term is
added in addition to the physical and geometric factors:

$$\psi_{12} = \psi_{\text{geom}} \psi_{\text{phys}} \psi_{\text{fresh}}.$$  \hspace{1cm} (5.10)

This freshness success factor considers the mass fraction of binder polymer (hydroxypropyl cellulose (HPC) throughout this dissertation) relative to the entire binder content (i.e., binder polymer and residual water). The implementation of this freshness factor is modeled after the physical factor in that coagulation is permitted until a critical binder content, $B_c^*$ is reached:

$$\psi_{\text{fresh}} = \begin{cases} 
1 & \text{if } B_c \text{ of either granule } < B_c^* \\
0 & \text{if } B_c \text{ of both granules } \geq B_c^*
\end{cases}$$  \hspace{1cm} (5.11)

At this point, too little water is considered to be present to dissipate the inertial forces of collision. This critical binder content is binder specific and must be determined experimentally. In this study, a critical binder content of 0.75 is used.

### 5.2.3 Evaporation Rate

The evaporation rate for this system is derived from a form of the mass flux equation:

$$n_A = k \rho_{\text{air}} (x_{\text{sat}} - x_{\text{out}})$$  \hspace{1cm} (5.12)

where $k$ is the mass transfer coefficient in m/s, $\rho_{\text{air}}$ is the density of the wet air in the granulator in kg/m$^3$, $x_{\text{sat}}$ is the mass fraction of water to wet air at saturation conditions in the granulator, and $x_{\text{out}}$ is the mass fraction of water to wet air at ambient conditions in the granulator. Because the source of the water vapor in this equation is a polymeric binder solution and not pure water, the $x_{\text{sat}}$ term is adjusted by an empirical correction factor $\phi$. This parameter is analogous to Raoult’s Law for the calculation of the vapor pressure of a solution and serves to lower the evaporation driving force. We use a value for $\phi$ that is equal to the initial mass fraction of water in a binder droplet: 0.85. The updated mass flux equation is thus:

$$n_A = k \rho_{\text{air}} (\phi x_{\text{sat}} - x_{\text{out}})$$  \hspace{1cm} (5.13)
The mass transfer coefficient is assumed to be constant and calculated from an approximation of the Ranz-Marshall correlation of the Sherwood number:

\[ k = 2 \frac{D_{\text{water}}}{d_p} \]  

(5.14)

where \( D_{\text{water}} \) is the diffusion coefficient of water vapor in air and \( d_p \) is the characteristic length taken to be the initial diameter of the particle. The diffusion coefficient is calculated using the following relationship [133]:

\[ D_{\text{water}} = 2.91 \times 10^{-5} \left( \frac{1}{P} \right)^{1.75} \frac{T_{\text{abs}}}{273.1} \]  

(5.15)

where \( P \) is the pressure in atmospheres (taken to be 1 atm) and \( T_{\text{abs}} \) is the absolute temperature in Kelvin. The diffusivity \( D_{\text{water}} \) has units of \( \text{m}^2/\text{s} \).

To obtain an overall evaporation rate for the entire granulator with units of \( \text{kg/s} \), the mass flux is multiplied by the total wet surface area of the granules, \( A_{p,\text{wet}} \):

\[ \dot{M}' = n_A A_{p,\text{wet}} = k A_{p,\text{wet}} \rho_{\text{air}} (\phi x_{\text{sat}} - x_{\text{out}}). \]  

(5.16)

The total wet surface area in the granulator is considered to be proportional to the total surface area of the granules and the moisture content of the bed denoted as \( x_{\text{bed}} \) (with units of kg water / kg dry powder). This relationship is given by the equation:

\[ A_{p,\text{wet}} = \omega x_{\text{bed}} A_{p,\text{total}} \]  

(5.17)

where \( \omega \) is a constant of proportionality. Substituting this relationship into Eq. 5.16, and noting that \( \rho_{\text{air}} x_{\text{text}} \) corresponds to the concentration of moisture in the granulator, yields

\[ \dot{M}' = k \omega x_{\text{bed}} A_{p,\text{total}} (\phi C_{w}^{\text{sat}} - C_w). \]  

(5.18)

with units of \( \text{kg/s} \). Converting this evaporation rate into a specific rate relative to the granulator volume (with units of \( \text{kg/m}^3\text{s} \)) yields:

\[ \dot{M}_{\text{evap}} = k \omega x_{\text{bed}} \frac{A_{p,\text{total}}}{V_R} (\phi C_{w}^{\text{sat}} - C_w). \]  

(5.19)

Here, we note that the average total surface area of the granules in the granulator
may be calculated from
\[
\frac{A_{p,\text{total}}}{C_pV_R} = \frac{A_{p,\text{total}}}{C_pV_R}
\] (5.20)

where \(C_pV_R\) is the total number of particles in the granulator. Upon multiplying Eq. 5.19 by \(C_p/C_p\) and utilizing the equation for \(A_{p,\text{total}}\), the final dimensional evaporation rate equation is obtained:
\[
\dot{M}_{\text{evap}} = k_\omega x_{\text{bed}} C_p A_{p,\text{total}} (\phi C_{w}^{\text{sat}} - C_w).
\] (5.21)

This equation is very similar in form, and in fact may be derived from, an analogous expression for the evaporation of moisture from food materials using a “reaction engineering model” [134]. In dimensionless form, this equation is written as
\[
\dot{m}_{\text{evap}} = \tau \frac{C_{\tau_{\text{evap}}}}{\tau_{\text{evap}}} x_{\text{bed}} C_p \frac{A_{p,\text{total}}}{A_{p,\text{total}}} (\phi C_{w}^{\text{sat}} - C_w).
\] (5.22)

A detailed discussion on the derivation of the dimensionless evaporation rate and associated parameters may be found in section 5.4.

As with many mass transfer problems, the key driving force during evaporation is the concentration difference between ambient and saturation moisture conditions within the granulator. The ambient conditions are calculated at each time step in the simulation. Because isothermal conditions are assumed, the experimentalist must provide the temperature of the air leaving the granulator in order to determine the saturation conditions. The following form of the Antoine equation is then used to calculate the saturation pressure of the water in units of Pascal:
\[
\ln P = A - \frac{B}{T_{\text{abs}} + C}
\] (5.23)

where \(A=21.8261\), \(B=3130.7984\), \(C=70.657\), and \(T_{\text{abs}}\) is the absolute temperature in Kelvin. From this pressure, the mol ratio of water to dry air can be obtained using the relationship
\[
\frac{P_{\text{water}}}{P_{\text{dry air}}} = \frac{n_{\text{water}}}{n_{\text{dry air}}}
\] (5.24)
The mass fraction of water to wet air is now obtained from

\[ x_{\text{sat}} = \frac{\text{mass}_{\text{water}}}{\text{mass}_{\text{water}} + \text{mass}_{\text{dry air}}} \]  

using molecular weights of 18 and 29 for water and air, respectively. It should be noted that this same procedure is used to calculate the moisture content of the incoming air from the inlet dewpoint temperature.

### 5.3 Development of Constant-Number Monte Carlo with Evaporation

The Monte Carlo model presented in Chapter 4 serves as the platform from which the Monte Carlo with evaporation model is built. The basic algorithm of the Monte Carlo process remains unchanged:

1. Calculate the probability of each event.
2. Choose the event to occur via a uniform random number generator.
3. Implement the selected event.
4. Update all metrics (mass concentration, number concentration, time advancement, etc.).

However, additional information is required to describe the two new processes considered: evaporation and flow of moisture out of the system. Details regarding the implementation of constant-number Monte Carlo are provided in the following subsections.

#### 5.3.1 Event Rates and Probabilities

A Monte Carlo simulation is based on random realization of events that alter the state of the simulation box. Each event has the potential to alter the number of particles, the mass of the particles, or the moisture content in the granulator; these changes will be denoted as \([\delta z]_i\), \([\delta x_p]_i\), and \([\delta x_w]_i\) for each event \(i\) defined below.
A summary of all rates and associated particle or moisture changes is provided in Table 5.2.

5.3.1.1 Coagulation Event

A coagulation event is defined as the successful coagulation of a particle pair in the simulation box. The dimensionless rate of coagulation is as defined in Chapter 2:

$$r_{\text{coag}} = \frac{\langle k_{12} \rangle c_p^2}{2}$$

(5.26)

Coagulation changes the number of particles by 1,

$$[\delta z]_{\text{coag}} = -1$$

(5.27)

but has no effect on the mass or the moisture content in the granulator:

$$[\delta x_p]_{\text{coag}} = 0$$

(5.28)

$$[\delta x_w]_{\text{coag}} = 0$$

(5.29)

5.3.1.2 Evaporation Event

An evaporation event is defined as the evaporation of a dimensionless mass $w$ of water from the particles. If all particles are considered dry, i.e., all particles contain a water content less than $w$, then the evaporation rate is set equal to zero. Otherwise, the rate of the evaporation event is given as

$$r_{\text{evap}} = \frac{\dot{m}_{\text{evap}}}{w}$$

(5.30)

An evaporation event has no effect on the number of particles,

$$[\delta z]_{\text{evap}} = 0,$$

(5.31)

but causes the transfer of mass $w$ from the particles to the gas:

$$[\delta x_p]_{\text{evap}} = -w$$

(5.32)

$$[\delta x_w]_{\text{evap}} = +w$$

(5.33)
A detailed discussion on the derivation of the dimensionless Monte Carlo evaporation event rate may be found in section 5.4.

5.3.1.3 Flow In Event

The inlet (flow-in) event is defined as the introduction of a dimensionless volume $\epsilon_{in}$ of flow into the simulation box. The rate of this event is

$$ r_{in} = \frac{1}{\tau_{in} \epsilon_{in}} \quad (5.34) $$

The number of particles (binder drops), $n$, that is carried by this stream is a function of the volume of flow into the granulator and the number of binder droplets that enter the granulator per unit volume of flow

$$ n = \epsilon_{in} c_{n0}. \quad (5.35) $$

The change in the number of particles is thus

$$ [\delta z]_{in} = n, \quad (5.36) $$

The total mass of droplets that enter the granulator per event is

$$ [\delta x_p]_{in} = n v_b \quad (5.37) $$

and the amount of moisture that accompanies this flow is

$$ [\delta x_w]_{in} = \epsilon_{in} c_{w0} = n \frac{c_{w0}}{c_{n0}} \quad (5.38) $$

The last relationship is obtained by substituting $\epsilon_{in}$ in terms of $n$ using Eq. (5.35). A detailed discussion on the derivation of the dimensionless Monte Carlo flow in event rate may be found in section 5.4.
5.3.1.4 Flow Out Event

The outlet flow (flow-out) event is defined as the removal of a dimensionless volume $\epsilon_{\text{out}}$ of flow from the simulation box. The rate of the event is

$$r_{\text{out}} = \frac{1}{\tau_{\text{out}}\epsilon_{\text{out}}} \quad (5.39)$$

The amount of moisture that is carried out of the simulation box is

$$[\delta x_w]_{\text{out}} = -\epsilon_{\text{out}}c_w \quad (5.40)$$

There is no effect on either the number or mass concentration of particles:

$$[\delta z]_{\text{out}} = 0 \quad (5.41)$$

$$[\delta x_p]_{\text{out}} = 0 \quad (5.42)$$

A detailed discussion on the derivation of the dimensionless Monte Carlo flow out event rate may be viewed in the “Dimensionless Monte Carlo Flow In Event Rate Derivation” portion of section 5.4.

5.3.1.5 Event Probability

The four possible events that can alter the contents of the simulation box are: flow in, flow out, evaporation, and coagulation. Once all event rates are calculated, the probability of each event $i$ may also be calculated:

$$P_i = \frac{r_i}{\sum_i r_i} \quad (5.43)$$

A uniform random number generator is used to choose which event to implement at each Monte Carlo step.

5.3.2 Event Implementation

Details on the implementation of events and their effect on the distribution of particles are briefly discussed below.
Table 5.2: Key constant-number Monte Carlo equations for granulation with evaporation

<table>
<thead>
<tr>
<th></th>
<th>Coagulation</th>
<th>Evaporation</th>
<th>Flow In</th>
<th>Flow Out</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rate</td>
<td>( r_{\text{coag}} = \frac{\langle k_{12} \rangle c^2_p}{2} )</td>
<td>( r_{\text{evap}} = \frac{\dot{m}_{\text{evap}}}{w} )</td>
<td>( r_{\text{in}} = \frac{c_{n0}}{\tau_{\text{in}} n} )</td>
<td>( r_{\text{out}} = \frac{1}{\tau_{\text{out}} c_{\text{out}}} )</td>
</tr>
<tr>
<td>( \tau_{\text{event}} )</td>
<td>( \tau_{\text{coag}} = \frac{1}{K_0 C_0} )</td>
<td>( \tau_{\text{evap}} = \frac{1}{A_b C_0 \omega k} )</td>
<td>( \tau_{\text{in}} = \frac{V_R}{Q_{\text{in}} \tau_C} )</td>
<td>( \tau_{\text{out}} = \frac{V_R}{Q_{\text{out}} \tau_C} )</td>
</tr>
<tr>
<td>Probability</td>
<td>( P_{\text{coag}} = \frac{r_{\text{coag}}}{\sum_i r_i} )</td>
<td>( P_{\text{evap}} = \frac{r_{\text{evap}}}{\sum_i r_i} )</td>
<td>( P_{\text{in}} = \frac{r_{\text{in}}}{\sum_i r_i} )</td>
<td>( P_{\text{out}} = \frac{r_{\text{out}}}{\sum_i r_i} )</td>
</tr>
<tr>
<td>( \delta[z]_i )</td>
<td>-1</td>
<td>0</td>
<td>( n )</td>
<td>0</td>
</tr>
<tr>
<td>( \delta[x_p]_i )</td>
<td>0</td>
<td>( -w )</td>
<td>( n v_b )</td>
<td>0</td>
</tr>
<tr>
<td>( \delta[x_w]_i )</td>
<td>0</td>
<td>( +w )</td>
<td>( n \frac{c_{w0}}{c_{n0}} = e_{\text{in}} c_{w0} )</td>
<td>( -c_{\text{out}} c_w )</td>
</tr>
</tbody>
</table>

Quantity Equation

- **Moisture mass conc.**  \( c_w = c_{w,\text{old}} + \frac{[\delta x_w]_i}{N \bar{v}} c_m \)
- **Particle mass conc.**  \( c_m = c_{m,\text{old}} + \frac{[\delta x_p]_i}{N \bar{v}} c_m \)
- **Particle number conc.**  \( c_p = \frac{c_m}{\bar{v}} \)
- **Time between events**  \( \Delta \tau = \frac{c_p}{N \sum_i r_i} \frac{1}{\bar{v}} \)

**Coagulation:** A coagulation event is implemented as usual: pairs are picked until the condition

\[
\text{rnd} \leq \frac{k_{12}}{\max\{k_{12}\}}
\]

is satisfied (here, \( k_{12} \) is the dimensionless kernel associated with particles 1 and 2, \( \max\{k_{12}\} \) is the maximum \( k_{12} \) between any two particles, and \( \text{rnd} \) is...
a random number). Following coagulation, a random particle is picked to fill the vacancy caused by coagulation.

**Evaporation:** A particle is picked and its moisture content is decreased by \( w \). The selected particle must have sufficient water content for an evaporation event to occur (i.e., the particle has a water content higher than \( w \) initially).

**Flow in:** During a flow in event, \( n \) binder particles are introduced into the simulation box. An equal number of particles are deleted to keep the sample size constant.

**Flow out** No effect on the particle sample.

### 5.3.3 Concentrations

When evaluating granulation with continuous binder addition and evaporation, three concentrations must be tracked: moisture mass concentration, particle mass concentration, and particle number concentration. An event generally changes the total mass in the simulation box by some amount \( \delta y_i \). Total mass refers to the mass of all species that are tracked by the simulation; here, this term includes the mass of particles plus the mass of moisture. To translate a change of mass in the simulation box into a corresponding change in mass concentration, it is noted that the mass of particles in the simulation box, \( M \), at any point of the simulation represents a mass concentration \( c_m \) of particles. Accordingly, a change in mass \( \delta y' \) in the simulation box represents a change in mass concentration \( \delta c' \) according to the similarity relationship,

\[
\frac{\delta c'}{c_m} = \frac{\delta y'}{N\bar{v}}
\]

where \( N\bar{v} = M \) is the total particle mass in the simulation box. Equation (5.44) provides the connection between discrete changes due to MC events and corresponding changes in the mass concentrations. Accordingly, the change in \( c_m \) and in \( c_w \) following event \( i \) is

\[
\delta c_m = \left[ \frac{\delta x_{pl}}{N\bar{v}} \right] c_m
\]

\[
\delta c_w = \left[ \frac{\delta x_w}{N\bar{v}} \right] c_m
\]
The values of $[\delta x_p]_i$ and $[\delta x_w]_i$ are summarized in table 5.2. Note that $[\delta x_p]_i$ is equivalent to $[\delta v_{event}]_i$ in Chapters 2 and 4.

The number concentration is calculated through its relationship to the mass concentration:

$$c_p = \frac{c_m}{\bar{v}} \quad (5.47)$$

where $c_m$ is the mass concentration calculated above and $\bar{v}$ is the average mass of the simulation box.

### 5.3.4 Time Increment

Time is incremented according to the usual equation,

$$\delta \tau = \frac{c_p}{N} \frac{1}{\sum_i r_i} \quad (5.48)$$

and is calculated based on the current values of the number concentration, $c_p$, and event rates, $r_{coag}$, $r_{evap}$, $r_{in}$, and $r_{out}$. A summary of the applicable rate equations is given in Table 5.2.

### 5.4 Derivation of Equations

#### 5.4.1 Dimensionless Change in Number Concentration Derivation

The dimensional change in particle number concentration as a function of time is given by Equation 5.1:

$$\frac{dC_p}{dt} = -\frac{\langle K_{12} \rangle C_p^2}{2} + \frac{\dot{Q}}{V_R}.$$  

Using the following relationships,

$$C_p = c_p C_0$$

$$t = \tau \tau_C$$

$$\langle K_{12} \rangle = K_0 \langle k_{12} \rangle$$

$$\dot{Q} = Q_{in} C_{n0}$$
\[ Q_{in} = \frac{V_R}{\tau_{in}\tau_C} \]
\[ C_{n0} = c_{n0}C_0 \]

this equation may be rewritten as
\[
\frac{dc_pC_0}{d\tau_C} = -K_0 \left\langle k_{12} \right\rangle c_p^2 \frac{C_0^2}{2} + \frac{V_R}{\tau_{in}\tau_C} \frac{c_{n0}C_0}{V_R}
\]

which, after simplification, yields
\[
\frac{dc_p}{d\tau} = -K_0C_0\tau_C \frac{\left\langle k_{12} \right\rangle c_p^2}{2} + \frac{c_{n0}}{\tau_{in}}.
\]

Upon invoking the definition of \( \tau_C \),
\[ \tau_C = \frac{1}{K_0C_0}, \]

the equation may be written in final dimensionless form:
\[
\frac{dc_p}{d\tau} = -\frac{\left\langle k_{12} \right\rangle c_p^2}{2} + \frac{c_{n0}}{\tau_{in}}.
\]

### 5.4.2 Dimensionless Change in Particle Mass Concentration Derivation

The dimensional change in particle mass concentration as a function of time is given by Equation 5.2:
\[
\frac{dC_m}{dt} = \frac{\dot{Q}M_\rho_0}{V_R} - M_{evap}
\]

Using the following relationships,
\[ C_m = c_mC_0M_b \]
\[ t = \tau\tau_C \]
\[ \dot{Q} = Q_{in}C_{n0} \]
\[ Q_{in} = \frac{V_R}{\tau_{in}\tau_C} \]
\[ C_{n0} = c_{n0}C_0 \]
\[ M_{p0} = v_b M_b \]
\[ \dot{M}_{\text{evap}} = \frac{\dot{m}_{\text{evap}} C_0 M_b}{\tau_C} \]

This equation may be rewritten as
\[
\frac{d c_m C_0 M_b}{d \tau \tau_C} = \frac{V_R}{\tau_{in} \tau_C} c_{n0} v_b M_b - \frac{\dot{m}_{\text{evap}} C_0 M_b}{\tau_C}
\]

which, upon simplification, is
\[
\frac{d c_m}{d \tau} = \frac{c_{n0} v_b}{\tau_{in}} - \dot{m}_{\text{evap}}.
\]

### 5.4.3 Dimensionless Change in Moisture Mass Concentration Derivation

The dimensional change in moisture mass concentration as a function of time is given by Equation 5.3:
\[
\frac{d C_w}{d t} = \frac{Q_{\text{in}} C_{w0}}{V_R} - \frac{Q_{\text{out}} C_w}{V_R} + \dot{M}_{\text{evap}}
\]

Using the following relationships,
\[
C_w = c_w C_0 M_b \\
t = \tau \tau_C \\
Q_{\text{in}} = \frac{V_R}{\tau_{in} \tau_C} \\
C_{w0} = c_{w0} C_0 M_b \\
Q_{\text{out}} = \frac{V_R}{\tau_{out} \tau_C} \\
\dot{M}_{\text{evap}} = \frac{\dot{m}_{\text{evap}} C_0 M_b}{\tau_C}
\]

this equation may be rewritten as
\[
\frac{d c_m C_0 M_b}{d \tau \tau_C} = \frac{V_R}{\tau_{in} \tau_C} c_{w0} C_0 M_b - \frac{V_R}{\tau_{out} \tau_C} c_w C_0 M_b + \frac{\dot{m}_{\text{evap}} C_0 M_b}{\tau_C}
\]
which, upon simplification, is

\[ \frac{dc_w}{d\tau} = \frac{c_{w0}}{\tau_{in}} - \frac{c_w}{\tau_{out}} + \dot{m}_{evap} \]

### 5.4.4 Dimensionless Evaporation Rate Derivation

The dimensional evaporation rate expression given by Equation 5.21 is

\[ \dot{M}_{evap} = k\omega x_{bed} C_p A_{p,\text{total}} (\phi C_{w}^{\text{sat}} - C_w). \]

The total surface area of the granules is converted to a dimensionless area through the use of a characteristic area chosen to be the area of an initial (monodisperse) binder droplet, \( A_b = \pi d_b^2 \):

\[ \overline{A_{p,\text{total}}} = a_{p,\text{total}} A_b. \]

Similarly, the dimensionless concentrations are as defined in Table 5.1:

\[ C_w = c_w C_0 M_b \]
\[ C_{w}^{\text{sat}} = c_{w}^{\text{sat}} C_0 M_b \]
\[ C_p = c_p C_0 \]
\[ \dot{M}_{evap} = \frac{\dot{m}_{evap} C_0 M_b}{\tau_C} \]

Thus, the dimensional evaporation rate is transformed to

\[ \frac{\dot{m}_{evap} C_0 M_b}{\tau_C} = k\omega x_{bed} c_p C_0 \overline{a_{p,\text{total}}} A_b C_0 M_b (\phi C_{w}^{\text{sat}} - c_w). \]

The characteristic evaporation time is defined as

\[ \tau_{evap} = \frac{1}{A_b C_0 \omega k}. \]

Using this relation, the final form of the dimensionless evaporation rate is obtained:

\[ \dot{m}_{evap} = \frac{\tau_C}{\tau_{evap}} x_{bed} c_p \overline{a_{p,\text{total}}} (\phi C_{w}^{\text{sat}} - c_w). \]
5.4.5 Dimensionless Monte Carlo Evaporation Event Rate Derivation

The dimensional evaporation rate is already given as $\dot{M}_{\text{evap}}$ with units of kg/m$^3$/s. For the purposes of Monte Carlo, an evaporation event is defined as the removal of a mass $W$ of water from a granule. Thus, the dimensional rate of evaporation is:

$$R_{\text{evap}} = \frac{\dot{M}_{\text{evap}}}{W}$$

To convert this specific rate $R_i$ into a dimensionless rate $r_i$, the same procedure as in Chapter 2 is followed, namely to multiply by a characteristic volume and characteristic time (already defined as $1/C_0$ and $\tau_{\text{coag}}$, respectively). The dimensionless evaporation rate is written as:

$$r_{\text{evap}} = \frac{\dot{M}_{\text{evap}} \tau_C}{W C_0}.$$ 

Upon defining a dimensionless mass of water removed per evaporation event as $w = W/M_b$, where $M_b$ is the characteristic mass of a monodisperse binder droplet, and utilizing the relationship

$$\dot{M}_{\text{evap}} = \frac{\dot{m}_{\text{evap}} C_0 M_b}{\tau_C},$$

this equation is now

$$r_{\text{evap}} = \frac{\dot{m}_{\text{evap}} C_0 M_b}{\tau_C} \frac{1}{w M_b C_0} \tau_C.$$ 

Upon simplification, this equation becomes Eq. 5.30:

$$r_{\text{evap}} = \frac{\dot{m}_{\text{evap}}}{w}.$$ 

5.4.6 Dimensionless Monte Carlo Flow In Event Rate Derivation

The dimensional rate of flow into a granulator with volume $V_R$ is given as $Q_{\text{in}}$ with units of m$^3$/s. For the purposes of Monte Carlo, an flow in event is defined as the introduction of a volume $V_{\text{in}}$ of air into the granulator. Thus, the dimensional rate
of flow in, with units of events/m³s, is:

\[ R_{in} = \frac{Q_{in}}{V_{in} V_R} \]

As above, the dimensionless rate is given as

\[ r_{in} = R_{in} V_s t_s = \frac{Q_{in} \tau_C}{V_{in} V_R C_0}. \]

Upon defining a dimensionless volume of flow in as \( \epsilon_{in} = \frac{V_{in}}{V_s} \), where \( V_s \) is already defined as \( 1/C_0 \), and utilizing the relationship

\[ \tau_{in} = \frac{V_R}{Q_{in} \tau_C}, \]

this equation becomes Eq. 5.34

\[ r_{in} = \frac{1}{\tau_{in} \epsilon_{in}}. \]

An analogous procedure may be used to derive the dimensionless flow out rate.

## 5.5 Validation of cNMC algorithm

All simulations were conducted using FORTRAN and a computer with a 2.80 GHz processor and 1.00 GB of RAM. Before extensively analyzing the simulation results, we first validated the model. The validation of this complex mathematical model was accomplished in two stages. In the first stage, Monte Carlo simulations were compared with the solution of the governing equations via Mathematica for simplified cases. In the second stage, Monte Carlo simulation results were compared with the rigorous discrete method described in Chapter 3. Obtaining similar results for two separate solution techniques gives credence to the proper development of the Monte Carlo method.
Figure 5.2: Comparison of Monte Carlo particle mass concentration results with Mathematica results for a very high evaporation rate.

### 5.5.1 Comparison of Monte Carlo with Mathematica

The governing equations listed in section 5.2.1 may be solved in Mathematica for simplified cases. Because only two processes impact the particle number concentration (binder flow and coagulation), and this scenario was examined in Chapter 4, the number concentration calculation was not checked. However, the determination of both the particle mass and moisture mass concentrations warrant checking. For the purposes of this validation, the simplified conditions for comparison include the assumption of a constant kernel and a dimensionless evaporation rate of

$$\dot{m}_{\text{evap}} = \frac{\tau_C}{\tau_{\text{evap}}} c_{w}^{\text{sat}} - c_{w}.$$  \hfill (5.49)

A comparison of the dimensionless particle mass concentration as a function of time for a relatively high evaporation rate is given in Figure 5.2. Arbitrary values of $c_{n0} = 1.01557$, $v_{b} = 1$, $\tau_{\text{in}} = \tau_{\text{out}} = 16.7$, $c_{w}^{\text{sat}} = 0.01$, $\tau_C = 59.8802$, $\tau_{\text{evap}} = 1$, and $c_{w0} = 0.001$ were used. The selection of a low evaporation rate characteristic time ($\tau_{\text{evap}}$) corresponds to a relatively high evaporation rate. Similarly, the selection of a relatively large characteristic time for the flow of air into and out of the granulator
An evaluation of the dimensionless moisture mass concentration as a function of time for three evaporation rates is given in Figure 5.3. The arbitrary values described above were used, with the exception of $\tau_{\text{evap}}$. For the high evaporation case, the same $\tau_{\text{evap}}$ was used (see Figure 5.3a), but a moderate evaporation case with $\tau_{\text{evap}} = 10,000$ (Figure 5.3b) and a very low evaporation case with $\tau_{\text{evap}} = 1,000,000$ (Figure 5.3c) were also analyzed. In all cases, the Monte Carlo simulations match well with Mathematica. This consistency of the concentration profile comparisons serves to validate Monte Carlo for a simple model throughout the entire length of the granulation process.
Surface coverage, \( \eta \)
50 40 30 20 10 0
Time (min)
cNMC
RM analytical equation

Figure 5.4: Comparison of average binder surface coverage as calculated via Monte Carlo and rigorous discrete (RM) analytical equation.

5.5.2 Comparison of Monte Carlo with the Rigorous Discrete Method

As discussed in Chapter 3, the rigorous discrete method is extremely accurate at early granulation times (i.e., for low granule sizes), but is computationally expensive. However, in contrast to the simplified case described above, the rigorous method permits the use of more complicated kernels and evaporation rates. In this validation, the KTGF kernel is used, but without the newly created freshness success factor. In addition, a more complicated evaporation rate is used:

\[
\dot{m}_{\text{evap}} = \frac{\tau C}{\tau_{\text{evap}}} \eta \bar{a}_{p,\text{total}} (c_{w}^\text{sat} - c_{w}) c_{p}
\]  

(5.50)

where \( \bar{a}_{p,\text{total}} \) is the average wet area per granule. The parameter \( \eta \) corresponds to the average fraction of binder surface coverage. This equation is similar in form to Eq. 5.22. While several points of comparison were made between Monte Carlo and the rigorous method, the evolution of three parameters as a function of time are shown below. These parameters include \( \eta \) (see Figure 5.4), the evaporation rate (Figure 5.5), and the moisture content of the outlet air (5.6). In the rigorous
Figure 5.5: Comparison of evaporation rate as calculated via Monte Carlo and rigorous discrete (RM) method.

Figure 5.6: Comparison of outlet air moisture content as calculated via Monte Carlo and rigorous discrete (RM) method.
method, $\eta$ was calculated via an equation that related time to the continuous binder mass flow rate. The Monte Carlo $\eta$ was determined by averaging the binder content on all granules within the simulation box. Very good agreement is seen when comparing the two parameters as a function of time throughout the entire length of the simulation. This finding shows that both models were incorporating binder onto the granules at a similar rate. In the comparison of the outlet moisture content and evaporation rate, the Monte Carlo results exhibit a high degree of oscillation due to the large volumetric flow rate of air that was used. However, it is important to note that the fluctuations oscillate around the rigorous method solution for the early stages of the granulation. Because the rigorous method is considered valid at the beginning of the granulation, this strong comparison serves to validate Monte Carlo for a more complex evaporation model.

5.6 Results and Discussion

The experiments that form the basis for assessing the current granulation model are those of Rajniak et al. [3, 15], which examined the granulation profile of three excipients: microcrystalline cellulose (Avicel Ph102), mannitol (Pearlitol SD200) and dicalcium phosphate (A-TAB). The experiments are briefly reviewed below. Narrow sieve fractions (0.920 kg) of an excipient were granulated with a 15 wt.% aqueous solution of hydroxypropyl cellulose binder (HPC) in a bottom-spray Wurster type fluidized bed granulator. The parameters used to simulate the experimental conditions are summarized in Table 5.3.

5.6.1 Temperature and Granulation

Once the Monte Carlo method was validated, the impact of temperature on both the granule growth profile and the moisture content could be examined. The powder chosen for study was Mannitol though this choice is largely irrelevant; all powders would exhibit the same qualitative response to increasing granulation temperature. In the model, the outlet (saturation) temperature was varied from $T = 10 - 95^\circ C$ while all other parameters were as listed in Table 5.3 and held constant. Arbitrary values of $K_0 = 2.5 \cdot 10^{-14}$ and $\omega = 0.2$ were used. Because
Table 5.3: Simulation parameters for Monte Carlo granulation with evaporation

<table>
<thead>
<tr>
<th>Variable</th>
<th>Value</th>
<th>Units</th>
<th>Reference/Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>$N$</td>
<td>10000</td>
<td>—</td>
<td>Arbitrary selection</td>
</tr>
<tr>
<td>$n$</td>
<td>10</td>
<td>—</td>
<td>Optimized selection</td>
</tr>
<tr>
<td>$d_b$</td>
<td>60</td>
<td>μm</td>
<td>[3]</td>
</tr>
<tr>
<td>$\rho$</td>
<td>1021</td>
<td>kg/m$^3$</td>
<td>Given by experimentalist</td>
</tr>
<tr>
<td>$\mu$</td>
<td>0.762</td>
<td>kg/(m s)</td>
<td>[3]</td>
</tr>
<tr>
<td>$u_0$</td>
<td>5.5</td>
<td>m/s</td>
<td>Given by experimentalist</td>
</tr>
<tr>
<td>$V_R$</td>
<td>0.01</td>
<td>m$^3$</td>
<td>10.0 L granulator</td>
</tr>
<tr>
<td>$r_g$</td>
<td>60</td>
<td>μm</td>
<td>[2]</td>
</tr>
<tr>
<td>$Q_{\text{in}}$</td>
<td>0.028</td>
<td>m$^3$/s</td>
<td>Calculated value from [3]</td>
</tr>
<tr>
<td>$\rho_{\text{air}}$</td>
<td>1.184</td>
<td>kg/m$^3$</td>
<td>Given by experimentalist</td>
</tr>
<tr>
<td>$T_{\text{dewpt, in}}$</td>
<td>10</td>
<td>°C</td>
<td>Given by experimentalist</td>
</tr>
<tr>
<td>$T_{\text{sat}}$</td>
<td>25</td>
<td>°C</td>
<td>Given by experimentalist</td>
</tr>
<tr>
<td>$A_b$</td>
<td>$1.131 \cdot 10^{-8}$</td>
<td>m$^2$</td>
<td>Calculated value</td>
</tr>
<tr>
<td>$\omega$</td>
<td>variable</td>
<td>—</td>
<td>Optimized selection</td>
</tr>
<tr>
<td>$\phi$</td>
<td>0.85</td>
<td>—</td>
<td>Given by experimentalist</td>
</tr>
<tr>
<td>$w$</td>
<td>0.1</td>
<td>—</td>
<td>Optimized selection</td>
</tr>
<tr>
<td>$B^*_c$</td>
<td>0.75</td>
<td>—</td>
<td>Given by experimentalist</td>
</tr>
<tr>
<td>Mannitol $C_0$</td>
<td>$7.91 \cdot 10^{10}$</td>
<td>1/m$^3$</td>
<td>Calculated value</td>
</tr>
<tr>
<td>A-TAB $C_0$</td>
<td>$3.07 \cdot 10^{10}$</td>
<td>1/m$^3$</td>
<td>Calculated value</td>
</tr>
<tr>
<td>Avicel $C_0$</td>
<td>$3.30 \cdot 10^{10}$</td>
<td>1/m$^3$</td>
<td>Calculated value</td>
</tr>
<tr>
<td>Initial Mannitol $d_{43}$</td>
<td>129.6</td>
<td>μm</td>
<td>[15]</td>
</tr>
<tr>
<td>Initial A-TAB $d_{43}$</td>
<td>177.5</td>
<td>μm</td>
<td>[15]</td>
</tr>
<tr>
<td>Initial Avicel $d_{43}$</td>
<td>173.4</td>
<td>μm</td>
<td>[15]</td>
</tr>
<tr>
<td>Mannitol and A-TAB $\dot{Q}$</td>
<td>$1.693 \cdot 10^6$</td>
<td>1/s</td>
<td>Calculated value</td>
</tr>
<tr>
<td>Avicel $\dot{Q}$</td>
<td>$1.485 \cdot 10^6$</td>
<td>1/s</td>
<td>Calculated value</td>
</tr>
</tbody>
</table>

The inlet dewpoint temperature is 10°C, this outlet temperature corresponds to a case where no drying of the binder occurs. The granule growth profile was evaluated by tracking $d_{43}$ as a function of time. To track the moisture content, a parameter known as “Loss on Drying” (LOD) was used. LOD is a common experimental technique used in industry that calculates moisture content as the ratio of the moisture mass present in a sample to the dry mass of the sample (i.e., sample mass after all moisture has been removed in an oven). The $d_{43}$ and LOD profiles may be seen in Figure 5.7. In both cases, the profiles decrease as the
temperature increases. This result is to be expected, but has only recently been explained via modeling using microscale interactions [132]. The findings presented here are complementary to the work of Velarde [132] and emphasize the importance of incorporating moisture effects on fluidized bed granulation models. A higher temperature corresponds to a larger driving force in the evaporation rate due to the capacity of the air to hold more water. This increased evaporation rate more quickly dries out the binder to the point where the binder is no longer effective in promoting successful collisions. As shown in Figure 5.8, similar results are achieved when the temperature is held constant, but the evaporation rate constant $\omega$ is increased from values of 0.01 (extremely low evaporation rate) through 10.0. Such a scenario can be considered analogous to faster drying due to increasing fluidization velocity.

5.6.2 Binder Distribution

A benefit of Monte Carlo modeling is that an in-depth study of population distributions such as the binder content of granules and the moisture content of the binder can occur. In the following sections, the binder content as a function of temperature and the impact of Monte Carlo evaporation mass size is discussed.
5.6.2.1 Effect of Temperature

As already shown, the granulation temperature plays a significant role in the granule growth profile that is achieved. To further study the rationale behind why larger temperatures cause depressed growth profiles, the binder distribution was examined. This distribution may be subdivided into two parts. The first distribution includes the mass fraction of binder (both water and HPC) relative to the granule. However, superimposed within this distribution is another distribution for the mass fraction of HPC within the binder. An example of these two distributions is shown in Figure 5.9. In both Figure 5.9a and Figure 5.9b, the mass fraction axis is linear and ranges from 0 to 1, whereas the size axis is logarithmic and ranges from a $d_{43}$ of 0 to approximately 1,630 µm. It should be noted that the height of each bar is only relative to the other bars within that size class. Figure 5.9a corresponds to the amount of binder within a granule while Figure 5.9b corresponds to the amount of HPC within the binder. Both distributions are important for the promotion of a successful “sticking” collision. The binder mass fraction is critical because, depending on the particle morphology, a significant amount of binder may need to be present on the granule before coagulation can begin. However, the moisture content of the within the binder is also important; an HPC mass fraction that is above the critical binder content of 0.75 will impede granulation. Since we are currently examining the impact of temperature on gran-
ulation, and the binder moisture content helps determine the collision probability via the $\psi_{\text{fresh}}$ term within the kernel, only the HPC mass fraction distributions are shown in Figure 5.10 for granulation temperatures of $T = 25, 40, 60,$ and $95^\circ C$. At temperatures at or above $40^\circ C$, a significant percentage of the binder is considered dry (above the critical binder content of 0.75). This data matches nicely with the very low and similar $d_{43}$ growth profiles for these temperatures given in Figure 5.7a. The data also explains the low LOD profiles for these temperatures given in Figure 5.7b. Thus, the reason for a decrease in granule growth with an increase in temperature can be directly correlated with the drying of binder in the system.

5.6.2.2 Effect of Monte Carlo Evaporation Mass

An artifact of Monte Carlo simulations is that continuous processes must be discretized into events so that a single event can be implemented at each Monte Carlo time step. Thus, the choice of the “coarsening factor” (how much of a continuous process is permitted to occur when that processing event is selected) is important. If this factor is too large, simulation speed is gained at the expense of accuracy. On the other hand, if it is too small, great accuracy may be obtained through a painstakingly long process. In Chapter 4, the selection of the number of binder droplets to be introduced per binder flow in event was discussed. Here, the amount
Figure 5.10: Analysis of mass fraction content for HPC relative to total binder mass (HPC and water)

of mass evaporated from a granule at each time step is examined.

To evaluate the selection of a Monte Carlo evaporation mass, coarsening factor values of \( w = 0.1 \) and \( w = 0.01 \) were examined for Mannitol granulation times of 5, 15, and 45 minutes. Arbitrary values of \( K_0 = 2.5 \cdot 10^{-14} \) and \( \omega = 0.2 \) were used. All other model parameters were as listed in Table 5.3 and kept constant. During the simulation, when an evaporation event is chosen, the selected particle must have sufficient water content for an evaporation event to occur (i.e., the particle must have a water content higher than \( w \) initially). Because each binder droplet
contains an initial dimensionless water mass of 0.85, it is possible for a water mass of 0.05 to remain when $w = 0.1$. When this occurs, the simulation considers that particle to be “dry” until more binder is added. To evaluate the accuracy associated with this assumption, a comparison of the $d_{43}$ and LOD profiles for $w = 0.1$ and $w = 0.01$ at a granulation time of 15 minutes is given in Figure 5.11. In this figure, the $w = 0.01$ simulation is considered the most accurate because it always permits the evaporation of all moisture from a granule; no particles are considered artificially dry as in the $w = 0.1$ case. However, when evaluating the results, the $d_{43}$ and LOD profiles are extremely comparable. Thus, the use of $w = 0.1$ as the Monte Carlo evaporation mass is justified from an accuracy standpoint. The next item to consider is the computational time savings associated with using a higher value for $w$. A list of representative computational times to simulate Mannitol granulation over 5, 15, and 45 minutes is given in Table 5.4. For granulation
times of 5 and 15 minutes, the computational time increases significantly when \( \omega = 0.01 \) is used relative to when \( \omega = 0.1 \) is used. A simulation was not concluded for the case of \( \omega = 0.01 \) during a 45 minute granulation due to the lengthy time requirement. Nevertheless, the trend is clearly evident. Thus, for the purposes of this study, a Monte Carlo evaporation mass of 0.01 was used as the default value in all simulations. This value minimized the computation time while still maintaining the accuracy obtained by the use of a smaller coarsening factor.

### 5.6.3 Comparison with Experiment

The ultimate goal of the developed Monte Carlo model is to faithfully simulate the experimental data in regards to both granule growth and moisture content. Before comparing simulations with experiment, however, the impact of evaporation on granule mass and volume should be considered. Evaporation removes moisture from a granule, but experimental evidence suggests that shrinkage of the granule due to evaporation is minimal. As such, a convention is adopted that volume is conserved during the granulation process (during both coagulation and evaporation), but mass may be lost due to evaporation. This convention affects the KTGF kernel and the Stokes number since both parameters contain a mass-dependent term.

The dimensionless size-dependent portion of the KTGF kernel is given in Chapter 2 as

\[
Z_{12} = \left(\frac{v_1^{1/3} + v_2^{1/3}}{v_1^{1/3}}\right)^2 \left(\frac{1}{v_1} + \frac{1}{v_2}\right)^{1/2}
\]  

where the first term on the right is related to particle volume and the second term on the right corresponds to particle mass. Previously, \( v \) was defined as a dimensionless volume. When it was assumed that density remained constant, then this parameter also doubled as a dimensionless mass. Now, because granule density is assumed to change throughout granulation as a result of evaporation, a new parameter \( v_m \) is assigned as the dimensionless mass where

\[
v_{m,1} = \frac{\rho_1 V_1}{\rho_b V_b} = \rho_{\text{dim.}} v.
\]

The parameters \( \rho_1 \) and \( V_1 \) correspond to the density and volume of granule 1 whereas \( \rho_b \) and \( V_b \) correspond to the density and volume of an initial monodisperse
binder droplet. The $\rho_{\text{dim.}}$ term refers to a dimensionless density. To incorporate the effect of changing mass into the KTGF kernel, the size dependent portion of the kernel is modified to the following:

$$z_{12} = (v_1^{1/3} + v_2^{1/3})^2 \left( \frac{1}{v_{m,1}} + \frac{1}{v_{m,2}} \right)^{1/2}$$  \hfill (5.53)$$

A similar procedure is necessary for the Stokes number calculation introduced in Chapter 2. In the “Dimensionless Stokes Number Derivation” section of that chapter, the Stokes number was defined as

$$St = \frac{8\tilde{m}u_0}{3\pi\bar{d}^2}$$  \hfill (5.54)$$

where $\tilde{m}$ and $\tilde{d}$ are the reduced particle mass and diameter, $u_0$ is the velocity of the collision, and $\mu$ is the binder viscosity. The reduced particle mass and diameter are

$$\tilde{m} = \frac{m_1m_2}{m_1 + m_2}$$  \hfill (5.55)$$

and

$$\tilde{d} = \frac{d_1d_2}{d_1 + d_2}. \hfill (5.56)$$

Upon substituting $\rho_x\pi d_x^3/6$ for each mass term and simplifying, the Stokes number is rewritten as

$$St = \frac{4u_0}{9\mu} \frac{\rho_1\rho_2}{\rho_1d_1^3 + \rho_2d_2^3} d_1d_2(d_1 + d_2)^2$$  \hfill (5.57)$$

After converting the diameters to dimensionless volume, the Stokes number is

$$St = \frac{4u_0}{9\mu} \frac{\rho_1\rho_2}{\rho_1d_1^{1/3} + \rho_2d_2^{1/3}} v_1^{1/3}v_2^{1/3}(v_1^{1/3} + v_2^{1/3})^2$$  \hfill (5.58)$$

One final item to note is that the calculation of the concentrations listed in Table 5.2 should be conducted using the average particle mass $\bar{v}_m$ rather than $\bar{v}$ (now strictly the average particle volume). These modifications were not expected to produce major changes in the simulation; they were added for the sake of completeness.

Once the Monte Carlo model was updated to include the impact of changing
mass due to evaporation, simulations were conducted using the parameters specified in Table 5.3. The adjustable parameters $K_0$ and $\omega$ were modified as necessary to obtain good agreement with the experimental data. The experimental $d_{43}$ and LOD values are compared with simulation results for Mannitol and Avicel in Figures 5.12 and 5.13, respectively. The values for $K_0$ and $\omega$ that best matched the experiments are listed in Table 5.5. Only one experimental LOD measurement had been obtained for each experiment after granulation started; this data point was taken at the end of the granulation process (after all binder was added but be-
Table 5.5: Optimized values of $K_0$ and $\omega$

<table>
<thead>
<tr>
<th>Powder</th>
<th>$K_0$</th>
<th>$\omega$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mannitol</td>
<td>$13.5 \cdot 10^{-14}$ m$^3$/s</td>
<td>0.65</td>
</tr>
<tr>
<td>Avicel</td>
<td>$9.0 \cdot 10^{-14}$ m$^3$/s</td>
<td>0.40</td>
</tr>
</tbody>
</table>

fore final drying). As such, this point only served to provide an appropriate target point for where the simulation LOD should end. Thus, from an LOD standpoint, simulation runs were merely constrained to achieving the same endpoint as the experimental data rather than mimicking an entire profile.

Matching simulation results with experimental results for both the $d_{43}$ and LOD data was an iterative process since each adjustable parameter affected both profiles. Eventually, however, good agreement between the experimental $d_{43}$ and the simulation was obtained for both powders. This result is not entirely surprising since the same experimental $d_{43}$ values as in Chapter 4 served as the basis for comparing the granule growth profile here. However, the optimized $K_0$ value necessary to match the experimental data was expected to increase due to the presence of evaporation; this expectation was indeed proven to be the case.

5.7 Conclusions

The multidimensional population balance model developed in Chapter 4 was expanded to include the impact of moisture on granulation growth. Specifically, constant-number Monte Carlo was adapted so that moisture content can be tracked both within granules and in the ambient air within the granulator. In addition, the success factor was also modified to include a freshness factor; this factor prevents agglomeration from occurring when the binder on a granule has become too dry. Monte Carlo simulations were shown to correctly predict the expected depression of granule growth when granulation temperature is increased. cNMC simulations were also fitted to experimental data for two powders. Through the use of two adjustable parameters, Monte Carlo simulations can effectively mimic the experimental data for both the granule growth profile and the moisture content.
Chapter 6

Four-component Granulation

6.1 Chapter Overview

In this chapter, the culmination of this dissertation is presented. The ultimate goal of this work was to develop a mathematical model that captures the key parameters affecting multi-component granulation. The research presented here considers a four-component granulation that tracks the evolution of two different powders and both the solid and liquid (moisture) components of the binder; this addition brings the total number of components tracked during a simulation to four. The nuances associated with the incorporation of a second powder into the coagulation kernel, via the collision success factor, are discussed. Changes to the Monte Carlo simulation code (relative to what was presented in Chapter 5) are minimal. To provide a “proof of concept” for this type of approach, simulation results are compared with experimental results for a 50:50 blend (by weight) of two common pharmaceutical excipients. The simulation trends well with the experimental results in all categories in which a direct comparison is made: granule growth profile, final moisture content, and powder distribution across sieve cuts.

6.2 Background

As discussed in the introduction of this dissertation, the internal distribution of components within granules is an important area of study. While an ideal gran-
Figure 6.1: Analysis of experimental A-TAB mass fraction vs. sieve cut for granulation of a 50:50 wt% mixture of Mannitol and A-TAB. Figure reproduced from Rajniak et al. [3] with permission.

Granulation produces final granules that each have the exact same composition, this ideal is nearly impossible to achieve practically. Recent research indicates that a component may appear at above-average concentrations in some size classes and below average in others [3,16–18]. The cause for this inhomogeneity may be due to such factors as surface roughness [3] and the initial size difference between the two powders [18]. If one powder is relatively smooth, then less binder is required before granule growth can begin. However, a powder with a high degree of surface roughness will require more binder to “fill in the gaps” before granule growth occurs. Upon granulating these two powders together, one can envision a scenario where, especially at the early stages of granulation, granules consist of many smooth particles and few rough particles. Such results were presented by Rajniak et al. [3] for A-TAB (relatively rough) and Mannitol (relatively smooth) and shown in Figure 6.1. The relative distribution of A-TAB across the various sieve cuts (size classes) confirms the assumption of a preferential growth of granules rich in Mannitol in the early stages of the granulation. A model is desired that can mimic these results.
6.3 Kernel Development

To incorporate two powders into the mathematical model, one must consider how the composition of each granule impacts the coagulation kernel. Recall that the kernel used throughout this dissertation, the KTGF kernel, has a success factor that determines that rate of successful (i.e., sticking) collisions. It is within the success factor that composition-dependent effects are taken into account. In Chapter 5, this factor was defined as

\[ \psi_{12} = \psi_{geom} \psi_{phys} \psi_{fresh}. \]  

(6.1)

Each of the three terms in this parameter corresponds to one of three conditions that must be met for a collision to successfully produce a larger granule. The first term accounts for the probability that binder will be present on at least one of the two granule points that collide. The second term evaluates if enough binder is present with sufficient viscosity to dissipate the inertial forces associated with the collision. The third term is essentially a modification of the \( \psi_{phys} \) term in that it calculates if the binder is too dry to promote a successful collision. Any changes that must be made to each of these three terms due to the addition of a second powder is discussed below.

6.3.1 Changes to \( \psi_{geom} \)

The geometrical success factor is determined from [15]:

\[ \psi_{geom} = 1 - (1 - \eta_1)(1 - \eta_2). \]  

(6.2)

where the accessible binder fraction \( \eta_i \) for each granule \( i \) is defined as:

\[ \eta_i = \frac{1}{1 + e^{-b(y_i - c)}}. \]  

(6.3)

This equation is a correlation developed by Stepanek et al. where \( y_i \) is the volumetric binder to solid ratio of granule \( i \), while \( b \) and \( c \) are powder-specific empirical correlation parameters. The values to choose for \( b \) and \( c \) are straightforward when only one powder is considered. With the introduction of a second powder, a com-
posite value for $b$ and $c$ must be calculated. A logical way to implement this change is through the use of powder mass fractions within the granule (not including the mass contribution of binder). If $x_{A-TAB}$ and $x_{Mann}$ represent the mass fraction of A-TAB and Mannitol, respectively, then $b_i$ and $c_i$ for each granule $i$ may be calculated as follows:

$$b_i = x_{A-TAB}b_{A-TAB} + x_{Mann}b_{Mann}$$

(6.4)

and

$$c_i = x_{A-TAB}c_{A-TAB} + x_{Mann}c_{Mann}.$$  

(6.5)

This treatment allows a composition-specific range of values for $b$ and $c$. If the mass fraction of one of the powders is equal to zero, then the single powder value is recovered.

### 6.3.2 Changes to $\psi_{phys}$

The physical portion of the success factor is basically an on-off switch that either permits or prevents a successful collision based on the ratio of viscous to inertial forces. This factor is a function of the Stokes number and critical Stokes number between two colliding particles. The Stokes number is defined as in Chapter 5:

$$St = \frac{4 u_0}{9 \mu \rho_1 \rho_2 d_b} \frac{v_1^{1/3} v_2^{1/3} (v_1^{1/3} + v_2^{1/3})^2}{(\rho_1 v_1 + \rho_2 v_2)}. $$

(6.6)

All terms are as defined previously. Because no powder-specific terms are contained within the Stokes number, no changes to this equation are necessary.

The critical Stokes number is defined as

$$St^* = 2 \ln \frac{\lambda_{12}}{h_a}$$

(6.7)

and depends on the binder layer thickness between the two colliding particles, $\lambda_{12}$, and the asperity (roughness) of the primary particle surface, $h_a$. These values are powder specific and require modification. The binder layer thickness is calculated
from the relative displaced binder volume $\phi_i$ as per the relationship [2,15]

$$\lambda_{12} = r_g \phi_1^{1/3} + r_g \phi_2^{1/3}$$

(6.8)

where $r_g$ is the mean radius of gyration. The relative displaced binder volume $\phi_i$ corresponds to the volume of binder present on the surface of granule $i$ available for coagulation. It is also a function of the binder to solid ratio, $y_i$, and given by the equation

$$\phi_i = e^f(y_i-g)$$

(6.9)

where $f$ and $g$ are empirical correlation parameters again given by Stepanek et al. [2]. The same method is used to calculate granule-specific values for $f$ and $g$ as was done for $b$ and $c$ in the geometric portion of the success factor (i.e., via powder mass fractions within the granules):

$$f_i = x_{A-TAB}f_{A-TAB} + x_{Mann}f_{Mann}$$

(6.10)

and

$$g_i = x_{A-TAB}g_{A-TAB} + x_{Mann}g_{Mann}$$

(6.11)

Similarly, $h_a$ is calculated as:

$$h_a = x_{A-TAB}h_{A-TAB} + x_{Mann}h_{Mann}$$

(6.12)

because surface roughness is also powder specific.

### 6.3.3 Changes to $\psi^{\text{fresh}}$

The freshness portion of the success factor is another on-off switch that permits or prevents a successful collision based on the binder moisture content on the granules. This factor is simply a function of the actual binder content on the two colliding particles. Because no powder-specific terms are contained within the Stokes number, no changes to this equation are necessary.
6.4 Development of Constant-Number Monte Carlo

The four-component Monte Carlo model is nearly identical to the one used in Chapter 5. The only change is conceptually straightforward (though not as easy to implement) and involves the creation of new arrays within the simulation to track the volume and mass of a second powder. Each granule’s total volume $v$ or mass $v_m$ is then obtained from the sum of the contributions from powder 1, powder 2, binder polymer (HPC) and binder moisture (water). The event rates, concentration calculations, and calculation of time advancement are as written in Table 5.2 on page 87. Keep in mind that the concentrations should be conducted using the average particle mass $\bar{v}_m$ rather than $\bar{v}$. Another important point to remember, however, is that the average kernel used in the calculation of the coagulation rate is determined via the use of the modified collision success factor described above.

6.5 Results and Discussion

To test the validity of the four-component Monte Carlo method, simulations were run and compared with the results presented in Rajniak et al. [3]. In that publication, it can be seen that the granule growth profile for the Mannitol-A-TAB 50:50 blend is qualitatively very similar to the pure Mannitol granule growth profile throughout the experiment. This type of behavior would be expected in a system where a large percentage of the initial particles were Mannitol. As such, preliminary simulations were conducted with this scenario in mind.

Simulations were conducted using FORTRAN and a computer with a 2.80 GHz processor and 1.00 GB of RAM. Simulation parameters are listed in Table 6.1. The $\rho$ value in the table refers to the density of the powders and initial density of the binder solution. The starting particle size for each of the powders was that used in the pure powder studies: initial A-TAB powder $d_{43}$ of 177.5 $\mu$m and initial Mannitol powder $d_{43}$ of 129.6 $\mu$m. Because the powder densities are equivalent, this size discrepancy requires a 50:50 mixture by mass to have 71.98% Mannitol and 28.02% A-TAB particles on a number basis. Thus, out of 10,000 particles initially, the simulation box contained 7198 Mannitol particles and 2802 A-TAB particles. The adjustable parameters $K_0$ and $\omega$ were modified as necessary to
Table 6.1: Simulation parameters for Monte Carlo granulation with evaporation

<table>
<thead>
<tr>
<th>Variable</th>
<th>Value</th>
<th>Units</th>
<th>Reference/Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>$N$</td>
<td>10000</td>
<td>—</td>
<td>Arbitrary selection</td>
</tr>
<tr>
<td>$n$</td>
<td>10</td>
<td>—</td>
<td>Optimized selection</td>
</tr>
<tr>
<td>$d_b$</td>
<td>60</td>
<td>µm</td>
<td>[3]</td>
</tr>
<tr>
<td>$\rho$</td>
<td>1021</td>
<td>kg/m$^3$</td>
<td>Given by experimentalist</td>
</tr>
<tr>
<td>$\mu$</td>
<td>0.762</td>
<td>kg/(m s)</td>
<td>[3]</td>
</tr>
<tr>
<td>$u_0$</td>
<td>5.5</td>
<td>m/s</td>
<td>Given by experimentalist</td>
</tr>
<tr>
<td>$V_R$</td>
<td>0.01</td>
<td>m$^3$</td>
<td>10.0 L granulator</td>
</tr>
<tr>
<td>$r_g$</td>
<td>60</td>
<td>µm</td>
<td>[2]</td>
</tr>
<tr>
<td>$Q_{in}$</td>
<td>0.028</td>
<td>m$^3$/s</td>
<td>Calculated value from [3]</td>
</tr>
<tr>
<td>$\rho_{air}$</td>
<td>1.184</td>
<td>kg/m$^3$</td>
<td>Given by experimentalist</td>
</tr>
<tr>
<td>$T_{dewpt, in}$</td>
<td>10</td>
<td>°C</td>
<td>Given by experimentalist</td>
</tr>
<tr>
<td>$T_{sat}$</td>
<td>25</td>
<td>°C</td>
<td>Given by experimentalist</td>
</tr>
<tr>
<td>$A_b$</td>
<td>1.131 $\cdot$ $10^{-8}$</td>
<td>m$^2$</td>
<td>Calculated value</td>
</tr>
<tr>
<td>$\omega$</td>
<td>variable</td>
<td>—</td>
<td>Optimized selection</td>
</tr>
<tr>
<td>$\phi$</td>
<td>0.85</td>
<td>—</td>
<td>Given by experimentalist</td>
</tr>
<tr>
<td>$w$</td>
<td>0.1</td>
<td>—</td>
<td>Optimized selection</td>
</tr>
<tr>
<td>$B_c^*$</td>
<td>0.75</td>
<td>—</td>
<td>Given by experimentalist</td>
</tr>
</tbody>
</table>

A-TAB - Mannitol $C_0$ | 5.49 $\cdot$ $10^{10}$ | 1/m$^3$ | Calculated value |
Initial Mannitol $d_{43}$ | 129.6 | µm | [15] |
Initial A-TAB $d_{43}$ | 177.5 | µm | [15] |
$\dot{Q}$ | 1.693 $\cdot$ $10^6$ | 1/s | Calculated value |

obtain good agreement with the experimental data.

The experimental granule growth profile and final LOD value are compared with simulation results in Figure 6.2. Good agreement between experimental and simulation data is obtained for $K_0$ and $\omega$ values of 9.00-$10^{-14}$ and 0.4, respectively. However, it is also desired to match the distribution of A-TAB across size ranges. In the experimental data presented in Figure 6.1, granules less than 180 µm contained more than 50 weight % at 30 minutes (about halfway) into the simulation whereas granules greater than 250 µm contained significantly less than 50 weight % A-TAB. By the end of the simulation, all granules approached a more consistent value, though granules less than 250 µm still contained more A-TAB than did granules greater than 250 µm. The sieve cut analysis associated with the
Figure 6.2: Comparison of Monte Carlo simulations with experimental data for granule growth profile and moisture content of 50:50 wt% Mannitol and A-TAB.

Figure 6.3: Analysis of simulation A-TAB mass fraction vs. sieve cut for granulation of a 50:50 wt% mixture of Mannitol and A-TAB. The red line indicates the ideal granule composition of 50 weight % A-TAB across all sizes.

above simulation is shown in Figure 6.3. The simulation A-TAB mass fractions do not exactly match the experimental results, but the same qualitative trends are evident. In the simulation, as in the experiment, the smaller sieve cuts have a disproportionately large A-TAB content relative to the larger sieve cuts at 30 minutes into the granulation. At the end of the granulation, the larger simulation granules have increased their A-TAB concentration and approach the ideal granule mass.
Figure 6.4: Analysis of A-TAB mass fraction and HPC mass fraction as a function of size after 15 minutes of granulating a 50:50 wt% mixture of Mannitol and A-TAB.

Fraction denoted by the red line in the figure. However, simulation results indicate that the A-TAB mass fraction in the 180-250 µm range remains high even after granulation is complete. These results have not yet been optimized and warrant further study to understand the exact cause for such discrepancies.

One tool available in Monte Carlo that is not readily available to the experimentalist is a detailed analysis of 3D distributions as a function of size. As such, a detailed analysis can be performed on the distribution of A-TAB mass fraction across various sieve cuts and the distribution of binder moisture as a function of size. Graphs of such 3D distributions at granulation times of 15 minutes, 30 minutes, 45 minutes, and 55 minutes (the end of granulation) are shown in Figures 6.4, 6.5, 6.6, and 6.7. In these figures, the mass fraction axis is linear and ranges from 0 to 1, whereas the size axis is logarithmic and ranges from a \( d_{43} \) of 0 to approximately 1,630 µm. It should be noted that the height of each bar is only relative to the other bars within that size class. The graph on the left of each figure corresponds to the A-TAB mass fraction (relative to powders only) as a function of size. The graph on the right of each figure corresponds to the HPC binder mass fraction (relative to binder mass only) as a function of size. The trend of decreased A-TAB content in larger granule sizes that is suggested in the sieve cut
Figure 6.5: Analysis of A-TAB mass fraction and HPC mass fraction as a function of size after 30 minutes of granulating a 50:50 wt% mixture of Mannitol and A-TAB.

Figure 6.6: Analysis of A-TAB mass fraction and HPC mass fraction as a function of size after 45 minutes of granulating a 50:50 wt% mixture of Mannitol and A-TAB.
Figure 6.7: Analysis of A-TAB mass fraction and HPC mass fraction as a function of size after 55 minutes of granulating a 50:50 wt% mixture of Mannitol and A-TAB.

Analysis is clearly confirmed when examining the 3D distributions of A-TAB mass fraction versus size. At only 15 minutes into the simulation, the largest granules are clearly devoid of a significant amount of A-TAB (see Figure 6.4a). The HPC mass fraction plot at this same time shows that, though many of the granules are considered “dry” (because they supersede the critical binder content $B^*_c$ mass fraction of 0.75), a large spread of binder droplets with moisture still exists; thus, granulation can still proceed without incident. At a granulation time of 30 minutes (see Figure 6.5), the largest granules are still enriched with Mannitol and the smaller granules with A-TAB, but a wide distribution of A-TAB concentrations exist for many of the size classes. At the same time, the distribution of moisture in the binder narrows and becomes drier due to evaporation. As the granulation proceeds toward the end point (see Figures 6.6 and 6.7), the largest granules trend towards the ideal mass fraction of 50 weight % and begin to exhibit a narrower distribution. The granule moisture continues to generally become drier. At this latter point in the granulation, the granules are nearly always dry and can only successfully combine when fresh binder is introduced to the granules. The granulation slows significantly at this point as seen in the simulation $d_{43}$ profile in Figure 6.2. It is interesting to note that the largest granules remain more moist than the
bulk of the granules. However, these data points represent only a small percentage of granules.

6.6 Conclusions

The multidimensional population balance model developed in Chapter 5 was expanded to include the addition of a second powder to the granulation. This task was accomplished by modifying the powder-specific components of the collision success factor to take into account the mass fraction of each powder in the granule. Overall, the simulation results match favorably with the experimental results, especially in regard to the granule growth profile and expected moisture content endpoint. Slight discrepancies existed between simulation and experiment when examining A-TAB mass fraction as a function of sieve cut, but the same qualitative trends are still readily apparent. These preliminary results are promising and represent an area of continuing development.
Conclusions

7.1 Dissertation Review

It has been said that the “Holy Grail” of granulation modeling is the a priori prediction of growth rate and granule behavior [135]. The research presented in this dissertation is but a single quest on this journey. We started with a desire to develop a mathematical model that captures several key parameters affecting granulation. These parameters centered primarily on the interaction between particle morphology and binder content. Ultimately, a four-component granulation model was desired so that the granulation of two powders (e.g., API (Active pharmaceutical ingredient) and excipient) with an aqueous binder could be studied. This problem was, and still is, quite complex and required the use of a multicomponent population balance equations. The best way to approach the problem was to break it down into manageable pieces. Chapters 3-6 capture the four major milestones accomplished during this dissertation.

Because the solution of a multicomponent population balance equation must be accomplished via numerical solution techniques, Chapter 3 was devoted to the comparison of three common solution techniques. Specifically, a rigorous version of the discretization technique, the direct quadrature method of moments (DQ-MOM), and constant-number Monte Carlo were studied. The rigorous method is very detailed and exact deterministic technique; a population balance equation is written for every conceivable size class, within the limit of exhaustion for the computer programmer. However, this accuracy is limited to the early stages of
granulation due to its very large computational load. As such, this technique is limited to serving as a validation tool to confirm the correct development of other solution techniques. DQMOM is another deterministic solution technique that tracks the evolution of a populations’s moments rather than its entire distribution. The DQMOM method is computationally efficient, but gains its increased computational speed at the cost of a loss of detailed information about the population. It was found that, with proper selection of the base set of moments, DQMOM is remarkably accurate when the coagulation kernel has no composition dependence. However, when composition dependence is introduced, the accuracy of the method is variable and depends on the details of the initial distribution. The third technique is a stochastic method entitled constant-number Monte Carlo; this technique permits the computation of univariate and multivariate distributions over arbitrarily long times. This technique was determined to be accurate when studying both composition-independent and composition-dependent kernels. For this reason, constant-number Monte Carlo was determined to be the best choice when analyzing multi-dimensional, composition-dependent PBEs. The main drawback of constant-number Monte Carlo is that it is not well-suited for systems that involve space and time gradients; a well-mixed granulator assumption is required when implementing this technique.

In Chapter 4, the traditional batch-style Monte Carlo algorithm was adapted to simulate the continuous addition of a liquid binder into the simulation box. This adjustment necessitated the use of a two-component population balance model that tracked granule size and binder content. The goal here was to be able to reproduce the granule growth profiles of three powders that had three very different granulation growth profiles. It was found that Monte Carlo simulations were able to match well with the experimental data for all three growth profiles despite the use of only one adjustable parameter. It was also shown via binder distribution profiles obtained from Monte Carlo that the powder with the smoothest surface grew faster because less binder was necessary to promote successful collisions. Previously, this phenomena was predicted primarily from qualitative arguments.

Chapter 5 examined the impact of binder moisture and drying on the granules produced via fluid bed granulation. At this point, four processes (rate of binder flow/moisture into the system, rate of coagulation, rate of evaporation, and
rate of moisture flow out of the systems) and three granule components (powder size, binder polymer content, binder water content) were included in the Monte Carlo model. A new term $\psi_{\text{fresh}}$ was introduced to the collision success factor; this term considers the internal distribution of moisture within a granule and prevents coagulation between granules when the binder is too dry. With this parameter, Monte Carlo simulations were shown to correctly predict the expected depression of granule growth when granulation temperature is increased. In addition, when compared with experimental growth profiles and moisture content end points, Monte Carlo simulations compared very favorably despite the use of only two adjustable parameters.

The incorporation of a second powder into a four-component population balance model was presented in Chapter 6. This work, though preliminary, represents the culmination of this dissertation. With this model, the evolution of granule size, moisture content, and compositional distribution may be considered simultaneously. When simulations were compared with the experimental data for a representative blend of two powders, Monte Carlo results were able to closely match the granule growth profile and expected moisture content endpoint. The simulated compositional distribution was not an exact match, but did qualitatively provide the correct trends. The model developed within this chapter is an area of continuing development.

As noted in Chapter 1, the overarching goal of this dissertation was to develop a mathematical model that captures the key parameters affecting granulation. An effective pharmaceutical wet granulation model should include at least three components: (i) a population balance methodology that tracks the distribution and composition of all species of interest, (ii) a physically based description of agglomeration and breakage, and (iii) hydrodynamic modeling. While the third piece of the puzzle was beyond the current scope of work, significant strides were made in regards to the other two. It was shown that the constant-number Monte Carlo methodology can utilize a complex morphologically-based compositional-dependent coagulation kernel (criteria ii) and track the size evolution of two different powders while also considering the impact of moisture loss on the system (criteria i). Simulation results generally compared very favorably with experimental results. Thus, the development of the constant-number Monte Carlo model
contained within this dissertation helps to fill a critical need within the granulation community for a model that can provide a thorough analysis of multi-component granulation problems.

7.2 Future Work

Many directions may be taken that stem from the research presented within this dissertation. An area of “low hanging fruit” is in the area of FORTRAN code optimization. A desire of many who pursue granulation modeling is to have a robust model that is not too complex and may be comfortably run on any standard desktop or laptop computer. While many of the simulations run using the code developed here can be completed within a few hours to a day, the code is far from optimized in terms of computational efficiency. Any improvements to the simulation speed would greatly enhance the attractiveness of this model to the general public.

Another area of future work is to incorporate potential mechanisms for breakage into the granulation model. While coagulation is the dominant (and preferred) mechanism within a granulator, breakage may also occur. The various forms of breakage that occur inside a granulator include the rupture of liquid or solid bridges due to collisions with the walls or other particles. The potential for attrition, or the breakage of fine particles from a granule’s surface, is particularly problematic if one of the powders is preferentially present on the surface. The coupling of coagulation and breakage provides a more complete analysis of what is truly happening within most granulators, especially if the granulator under study is a high shear mixer.

A third area of future study involves modifying the current Monte Carlo model so that it can be coupled with hydrodynamic simulators such as FLUENT. Currently, the event-driven nature of constant-number Monte Carlo does not readily lend itself to interfacing with standard time-driven integrators used in the hydrodynamic simulators. The time-driven Monte Carlo process is a possibility, but suffers from other setbacks such as the potential for computational inefficiency if many time steps are necessary before a Monte Carlo event can occur. If Monte Carlo simulations can be adapted such that they can easily interface with hydrodynamic simulators, then a very robust granulation model can be constructed that tracks
the distribution of all species of interest via population balance considerations while simultaneously considering the impact of flow fields within the granulator.

Yet another modification that can be made to the model developed in this dissertation is the incorporation of a heat balance on the granulator. Such an adjustment would permit the modeler to consider non-isothermal effects on granulation as well as evaluate the heat transfer that occurs between the drying air and the granules during evaporation.

Other improvements that may be made to the model include studying the impact of having an initial distribution of particle and binder sizes and also considering the impact that hygroscopicity has on the granule growth profile. Highly hygroscopic materials dry the binder faster than a non-hygroscopic counterpart and theoretically lower the granulation growth profile as a result. The magnitude of such a shift is currently undefined.
FORTRAN code for Comparison of Different Methods

A.1 FORTRAN program overview

The FORTRAN code used in this dissertation is comprised of one main file, entitled “main.f95”, and several associated supporting files and subroutines. While a separate code was written for each of the four main chapters (chapters 3-6), the code used to create the results in Chapter 6 is presented in this appendix. However, with minor modification, the work contained in each of the preceding chapters may also be obtained. (In fact, the code presented herein has already been slightly modified to permit the generation of results from Chapter 5). The general flow of the code is as follows.

Global variables are defined in the file “particle_module.f95”. After these and the local variables are given, the main.f95 file calls for the initial distribution subroutine (“Initial_Distribution.f95”) to calculate the initial distribution of particles. It is within this section that the input files “Initial Dist Input1.txt” and “Initial Dist Input2.txt” (reference Appendix B) are accessed. Once the initial distribution of particles is created, some more initial parameters are defined or calculated. The simulation starts in earnest when the “event_selection.f95” subroutine is called. During this step, the probabilities for the various Monte Carlo events are calculated and one is selected. The coagulation of two particles is given in the “coag-
ulation.f95” file, the flow of binder into the system is shown in “binder_flow.f95” and the evaporation step is detailed in “evaporation.f95”. The calculation of system changes due to an event (e.g., change in particle number concentration and time) is performed by the subroutine given in “Calc_updates.f95”. Throughout the simulation, as well as at the end of the simulation, the “stats.f95” subroutine calculates other meaningful statistics such as the system LOD (loss on drying) and binder content.

Other supporting files include “KTGF_comp_depend_kernel.f95”, “KTGF_comp_depend_kernel2.f95”, “ran0.f95” and “optional_subs.f95”. The first two files calculate the kernel associated with any particle pairing. The difference between the two files is that KTGF_comp_depend_kernel.f95 accounts for particle-particle interactions whereas KTGF_comp_depend_kernel2.f95 calculates the kernel for particle-binder droplet interactions. The ran0.f95 file serves as a linchpin of the Monte Carlo process in that the random number generator is housed here. The final file, optional_subs.f95, is, as the name implies, an optional subroutine that can be used to view the particle array before and after a granulation event. Each of the 13 files listed above are reproduced in their entirety in the succeeding sections.
A.2 Main file

```fortran
program main

!***** Define global variables *****
use particle

!***** Define local variables *****
implicit none

integer i, j, k, kkk, lll  ! particle indices
integer runs  ! index for number of simulation runs
integer r  ! dummy index for simulation runs
integer ii, jj, kk, mm  ! disposable index
integer sieve_count  ! variable giving the # of sieves desired for the final distribution
integer coag_count, flowin_count, flowout_count, evap_count
real*8 maxloop  ! repetitions in loop
integer print_freq  ! printing frequency
real*8 kernel  ! kernel, by user-supplied function
real*8 lola  ! disposable variable
character(80) kernel_type  ! string that identifies the kernel
real*8 beta  ! solute to excipient size ratio = vA/vB
real*8 avg_v0  ! initial average mass of system

real*8 v43  ! parameter used to calculate x43
real*8 tau_c  ! characteristic coagulation time
real*8 tau_in  ! characteristic binder flow stream in time
real*8 tau_out  ! characteristic flow out time
real*8 tau_evap  ! characteristic evaporation time
real*8 N_spray  ! binder spray rate (1/s)
real*8 Q_in  ! volumetric flow of air in (m^3/s)
real*8 Q_out  ! volumetric flow of air out (m^3/s)

!!! real*8 run_dried_d43_sum(nmax)  ! gives array of dried d43 sums for multiple runs
!!! real*8 avg_dried_d43(nmax)  ! gives array of average dried d43 for multiple runs
!!! real*8 run_time_sum(nmax)  ! gives array of time sums for multiple runs
```
!!! real*8 avg_time(nmax) ! gives array of average time for multiple runs

real*8 Tdew_C
real*8 Tdew_K
real*8 Psat_dew
real*8 MW_h2o
real*8 MW_air
real*8 dew_mol_ratio
real*8 Tsat_C
real*8 Tsat_K
real*8 Psat_bed
real*8 sat_mol_ratio

integer*4 date_time(8)
character(len=10) date_time_zone(3)
character*80 FileName1, FileName2, FileName3, FileName4, FileName5, FileName6, FileName7, FileName8, FileName9, 
                FileName10, FileName11, FileName12, FileName13, FileName14, FileName15

real*8 ran0, rnd ! random number generator
integer iseed ! seed for rnd DO NOT SET iseed = 0

common/r/iseed
common/k/kernel_type

***** Prompt user to enter desired number of simulation runs *****
!! print *, 'Enter desired number of simulation runs in integer form
     (e.g., 10)'
!read *, runs
runs=1
write(*,*) "Simulation started"

***** Initiate values used to calculate averages over multiple runs *****
!!! maxdata=0

***** Open necessary output files *****
do 10 r=1, runs
! write (FileName1,100) r
! write (FileName2,200) r
write (FileName3,300) r
write (FileName4,400) r
write (FileName5,500) r
write (FileName6,600) r
write (FileName7,700) r
write (FileName8,800) r
write (FileName9,900) r
write (FileName10,150) r
write (FileName11,250) r
write (FileName12,350) r
write (FileName13,450) r
write (FileName15,650) r

! open (01, file = 'initial dist.txt') ! For storage of initial distribution

! open (11, file = FileName1)
! 100 format("excess solute",i3,".txt")
! open (11, file = 'excess solute.txt') ! For storage of Vavg and excess solute data

! open (21, file = FileName2)
! 200 format("segregation",i3,".txt")
! open (21, file = 'segregation.txt') ! For storage of segregation (or content heterogeneity) index data

open (31, file = FileName3)
! open (31, file = 'output.txt') ! For storage of individual granule mass and solute concentration
write(31,*) "FINAL PARTICLE DISTRIBUTION"
write(31,*) ""
300 format("output",i3,".txt")

! open (41, file = FileName4)
! 400 format("Particle mass vs time",i3,".txt")
! open (41, file = 'Particle mass vs time.txt') ! For storage of Vavg, tau, and SI (or content heterogeneity index)

open (51, file = FileName5) ! To check if any maximum kernel violations occurred
format("Max kernel violation._",i3,".txt")

open (71, file = 'Avg dried d43 data from runs.txt')

open (81, file = FileName6)

format("Particle Vavg and number conc vs time_.",i3,".txt")

! open (81, file = 'Particle Vavg and number conc vs time.txt') ! For storage of Vavg, # concentration, and time

open (91, file = FileName7)

format("cB cP and c_.",i3,".txt")

! open (91, file = 'cB cP and c.txt') ! For storage of Binder # conc, particle # conc, and total # conc

open (101, file = FileName8)

format("Granule output_.",i3,".txt")

! open (101, file = 'Granule output.txt') ! For storage of individual granule wet and dry mass

open (111, file = FileName9)

format("Granule mass only vs time_.",i3,".txt")

! open (111, file = 'Granule mass only vs time.txt') ! For storage of wet and dry gran. Vavg (not incl. binder droplets), tau, and LOD

open (121, file = FileName10)

format("Dried granule mass vs time_.",i3,".txt")

! open (121, file = 'Dried granule mass vs time.txt') ! For storage of dried granule Vavg and dried granule d43

! open (131, file = FileName11)

format("Dried granule output_.",i3,".txt")

! open (131, file = 'Dried granule output.txt')

open (141, file = FileName12)

! open (141, file = 'output mass.txt') ! For storage of individual granule mass, binder mass, and moisture mass

write(141,*) "FINAL PARTICLE DISTRIBUTION"

write(141,*) ""

format("output mass_.",i3,".txt")
open (151, file = FileName13)
format("Concentrations vs time_",i3,".txt")
! open (151, file = 'Concentrations vs time.txt') ! For storage of particle # conc, particle mass conc, moisture mass conc, and tau

open (171, file = FileName15)
format("List of variables_",i3,".txt")
! open (171, file = 'List of variables.txt') ! For storage of variables used in simulation

!**** Establish start time of analysis ****

call date_and_time(date_time_zone(1), date_time_zone(2), date_time_zone(3), date_time)

! write (01,*) 'Start date of analysis (month/day/year): ',date_time(2), '/', date_time(3), '/', date_time(1)
! write (01,*) 'Start time of analysis (hour:minute:second): ', date_time(5), ":", date_time(6), "", date_time(7)

! write (11,*) 'Start date of analysis (month/day/year): ',date_time(2), '/', date_time(3), '/', date_time(1)
! write (11,*) 'Start time of analysis (hour:minute:second): ', date_time(5), ":", date_time(6), "", date_time(7)
! write (11,*) ""
! write (11,*) " Vavg X—Excess Solute"

! write (21,*) 'Start date of analysis (month/day/year): ',date_time(2), '/', date_time(3), '/', date_time(1)
! write (21,*) 'Start time of analysis (hour:minute:second): ', date_time(5), ":", date_time(6), "", date_time(7)
! write (21,*) ""
! write (21,*) " Vavg X^2 SI Chi"

lola=kernel(1,1) ! call kernel to get kernel_type
write (31,*) "kernel: ", kernel_type
write (31,*) 'Start date of analysis (month/day/year): ',date_time(2), '/', date_time(3), '/', date_time(1)
write (31,*): 'Start time of analysis (hour:minute:second): ', date_time(5),":", date_time(6),":", date_time(7)

write (41,*): 'Start date of analysis (month/day/year): ', date_time(2), '/', date_time(3), '/', date_time(1)
write (41,*): 'Start time of analysis (hour:minute:second): ', date_time(5),":", date_time(6),":", date_time(7)
write (41,*): " Vavg Time SI"

write (141,*): "kernel: ", kernel_type

write (141,*): 'Start date of analysis (month/day/year): ', date_time(2), '/', date_time(3), '/', date_time(1)
write (141,*): 'Start time of analysis (hour:minute:second): ', date_time(5),":", date_time(6),":", date_time(7)

!!! write (71,*): 'Start date of analysis (month/day/year): ', date_time(2), '/', date_time(3), '/', date_time(1)
!!! write (71,*): 'Start time of analysis (hour:minute:second): ', date_time(5),":", date_time(6),":", date_time(7)
!!! write (71,*): "runs completed"
!!! write (71,*): ""

!!! write (71,*): "Dried d43 Time"

write (81,*): 'Start date of analysis (month/day/year): ', date_time(2), '/', date_time(3), '/', date_time(1)
write (81,*): 'Start time of analysis (hour:minute:second): ', date_time(5),":", date_time(6),":", date_time(7)

write (91,*): 'Start date of analysis (month/day/year): ', date_time(2), '/', date_time(3), '/', date_time(1)
write (91,*): 'Start time of analysis (hour:minute:second): ', date_time(5),":", date_time(6),":", date_time(7)

write (101,*): 'Start date of analysis (month/day/year): ', date_time(2), '/', date_time(3), '/', date_time(1)
write (101,*): 'Start time of analysis (hour:minute:second): ', date_time(5),":", date_time(6),":", date_time(7)
write (111,*) 'Start date of analysis (month/day/year): ', date_time (2), '/', date_time (3), '/', date_time (1)
write (111,*) 'Start time of analysis (hour:minute:second): ',
        date_time (5), ':', date_time (6), ':', date_time (7)
write (111,*) ""
write (111,*) " Wet Granule Vavg Time Dry Granule Vavg Wet LOD Dry LOD"

write (121,*) 'Start date of analysis (month/day/year): ', date_time (2), '/', date_time (3), '/', date_time (1)
write (121,*) 'Start time of analysis (hour:minute:second): ',
        date_time (5), ':', date_time (6), ':', date_time (7)
write (121,*) ""
write (121,*) " Dried Granule Vavg Time Dried Granule d43 Rate ratio (coag/binder)"

! write (131,*) 'Start date of analysis (month/day/year): ', date_time (2), '/', date_time (3), '/', date_time (1)
! write (131,*) 'Start time of analysis (hour:minute:second): ',
        date_time (5), ':', date_time (6), ':', date_time (7)
write (151,*) "kernel: ", kernel_type
write (151,*) 'Start date of analysis (month/day/year): ', date_time (2), '/', date_time (3), '/', date_time (1)
write (151,*) 'Start time of analysis (hour:minute:second): ',
        date_time (5), ':', date_time (6), ':', date_time (7)
write (171,*) 'Start date of analysis (month/day/year): ', date_time (2), '/', date_time (3), '/', date_time (1)
write (171,*) 'Start time of analysis (hour:minute:second): ',
        date_time (5), ':', date_time (6), ':', date_time (7)
write (171,*) ""

print *, 'Start date of analysis (month/day/year): ', date_time (2), '/'
        ', date_time (3), '/', date_time (1)
print *, 'Start time of analysis (hour:minute:second): ', date_time (5), ':', date_time (6), ':', date_time (7)
！”MAIN INPUT PARAMETERS HERE！”

! print *, 'Enter value for beta (ratio of v1/vB) in decimal form (e., 1.0).'
! read *, beta

!! print *, 'Enter number of powders to be studied (1 or 2):'
! read *, powder_number_flag

powder_number_flag=2
if(powder_number_flag.ne.1.and.powder_number_flag.ne.2) then
  print *, 'Please enter either 1 OR 2.'
  read *, powder_number_flag
end if
if(powder_number_flag.eq.1) then
  write (171,*), "Number of powders = 1"
elsif(powder_number_flag.eq.2) then
  write (171,*), "Number of powders = 2"
end if

!! print *, 'Enter value to define the powder to be studied (powder 1):'
!! print *, ' Avicel: Enter 1'
!! print *, ' A−TAB: Enter 2'
!! print *, ' Lactose: Enter 3'
!! print *, ' Mannitol: Enter 4'

! read *, powder_flag1

powder_flag1=4
if(powder_flag1.ne.1.and.powder_flag1.ne.2.and.powder_flag1.ne.3.
  and.powder_flag1.ne.4) then
  print *, 'Please enter either 1, 2, 3 OR 4.'
  read *, powder_flag1
end if
if(powder_flag1.eq.1) then
  write (171,*), "Powder 1 = Avicel"
elsif(powder_flag1.eq.2) then
  write (171,*), "Powder 1 = A−TAB"
elsif(powder_flag1.eq.3) then
  write (171,*), "Powder 1 = Lactose"
elsif(powder_flag1.eq.4) then
  write (171,*), "Powder 1 = Mannitol"
end if
!! print *, 'Enter value to define the powder to be studied (powder 2),'
!! print *, ' Avicel: Enter 1'
!! print *, ' A-TAB: Enter 2'
!! print *, ' Lactose: Enter 3'
!! print *, ' Mannitol: Enter 4'
read *, powder_flag2
powder_flag2=2
if(powder_flag2.ne.1.and.powder_flag2.ne.2.and.powder_flag2.ne.3.
   and.powder_flag2.ne.4) then
   print *, 'Please enter either 1, 2, 3 OR 4.'
   read *, powder_flag2
end if
if(powder_flag2.eq.1) then
   write (171,*) "Powder 2 = Avicel"
elseif(powder_flag2.eq.2) then
   write (171,*) "Powder 2 = A-TAB"
elseif(powder_flag2.eq.3) then
   write (171,*) "Powder 2 = Lactose"
elseif(powder_flag2.eq.4) then
   write (171,*) "Powder 2 = Mannitol"
end if

!! print *, 'Enter value that will determine if the initial
   monodisperse powder size is d32 or d43::'
!! print *, ' If initial powder size is based on d32: Enter 1'
!! print *, ' If initial powder size is based on d43: Enter 2'
!read *, d32_d43_flag
d32_d43_flag=2
if(d32_d43_flag.ne.1.and.d32_d43_flag.ne.2) then
   print *, 'Please enter either 1 OR 2.'
   read *, d32_d43_flag
end if
if(d32_d43_flag.eq.1) then
   write (171,*) "Initial powder size is based on d32"
elseif(d32_d43_flag.eq.2) then
   write (171,*) "Initial powder size is based on d43"
end if
!! print *, 'Enter value that will determine if d43 gamma is to be
 included in the kernel:'
!! print *, ' If d43 gamma is to be included: Enter 1'
!! print *, ' If d43 gamma is NOT to be included: Enter 2'
! read *, d43_gamma_flag

d43_gamma_flag=2
if(d43_gamma_flag.ne.1.and.d43_gamma_flag.ne.2) then
  print *, 'Please enter either 1 OR 2.'
  read *, d43_gamma_flag
end if
if(d43_gamma_flag.eq.1) then
  write (171,*) "d43 gamma term was included"
elseif(d43_gamma_flag.eq.2) then
  write (171,*) "d43 gamma term was NOT included"
end if

!print *, 'Enter value for d0 (diameter used to make sizes
dimensionless) in decimal form (e.g., 60.0d-6)'
! read *, d0
!d0=40.0d-6 ! Use for 10% HPC
!d0=60.0d-6 ! Use for 15% HPC
write (171,*) "d0 =", d0

!print *, 'Enter value for rho (powder density) in kg/m^3 and
decimal form (e.g., 1200.0)'
! read *, rho
!!rho=1200.0
rho=1021.0
write (171,*) "rho =", rho

!print *, 'Enter value for mu (binder viscosity) in cP and decimal
form (e.g., 762.0)'
! read *, mu_cp
mu_cp=762.0
mu=mu_cp/1000.0 !mu is in kg/m-s
write (171,*) "Binder viscosity (in cP) =", mu_cp

!print *, 'Enter value for u (collision velocity) in m/s and
decimal form (e.g., 0.25)'
! read *, u
u = 5.5

write (171, *) "u =", u

! print *, 'Enter value for vb (dimensionless binder mass) in decimal form (e.g., 1.0)'
! read *, vb

vb = 1.5d-1 ! Use for 10% HPC
vb = 1.0d-1 ! Use for 15% HPC
write (171, *) "vb =", vb

! print *, 'Enter value for wb (dimensionless moisture mass in binder) in decimal form (e.g., 1.0)'
! read *, wb

wb = 9.0d-1 ! Use for 10% HPC
wb = 8.5d-1 ! Use for 15% HPC
write (171, *) "wb =", wb

! print *, 'Enter value for wp (initial dimensionless moisture mass in powder) in decimal form (e.g., 1.0)'
! read *, wp

wp = 0.0
write (171, *) "wp =", wp

! print *, 'Enter value for LOD of powder 1 in decimal form (e.g., 0.005 for 0.5%)'
! read *, initial_LOD1
!! initial_LOD1 = 0.00564
!! write (171, *) "initial LOD of powder 1 =", initial_LOD1

! print *, 'Enter value for LOD of powder 2 in decimal form (e.g., 0.005 for 0.5%)'
! read *, initial_LOD2
!! initial_LOD2 = 0.0
!! write (171, *) "initial LOD of powder 2 =", initial_LOD2

! print *, 'Enter value for gamma (x43 exponent) in decimal form (e.g., 2.8)'
! read *, gamma

gamma = 2.8d0
write (171, *) "gamma =", gamma
!print *, 'Enter mean gyration radius (in um) in decimal form (e.g., 60.0)'
!read *, rg
rg=60.0
write (171,*) "rg =", rg

!print *, 'Enter percentage of A (powder) particles in decimal form (e.g., 0.50)'
!read *, fa
fa=1.0
write (171,*) "fa =", fa

!print *, 'Enter value for phi0 (initial mass fraction of solute) in decimal form (e.g., 0.5).'
!read *, phi0
!write (171,*) "phi0 =", phi0

!print *, 'Enter value for average v0 (initial average mass of system) in decimal form (e.g., 1.0).'
!read *, avg_v0
!write (171,*) "avg_v0 =", avg_v0

!print *, 'Enter number of binder particles to be added per event in integer form'
!read *, n_binder1
n_binder1=10.0
write (171,*) "n_binder1 =", n_binder1

!Establish second variable if < 1 binder droplet is to be added (e.g., 1 binder droplet per 10 binder selection events)
if(n_binder1.lt.1.0) then
  n_binder2=1
else
  n_binder2=n_binder1
end if
write(*,*) "n_binder1 =", n_binder1
write(*,*) "n_binder2 =", n_binder2
write (171,*) "n_binder2 =", n_binder2
Call for initial distribution subroutine that allows for a monodisperse, normal, or log-normal dist. of powder

```fortran
!****
call initial_dist

!print *, 'Enter particle–particle aggregation constant K_PP in
decimal form (e.g., 1.0d-13 m^3/s)'
!read *, K_PP
!!!K_PP=2.50d-14 !1.25d-13(A-TAB) !7.00d-14(Avicel) !2.50d-14(Mannitol)
write (171,*) "K_PP =", K_PP

!print *, 'Enter particle–binder aggregation constant K_PB in
decimal form (e.g., 1.0d-13 m^3/s)'
!read *, K_PB
!!!K_PB=2.50d-12
write (171,*) "K_PB =", K_PB

!print *, 'Enter C0 (initial # conc. of particles, #/m^3) in
decimal form (e.g., 1.67d11)'
!read *, C0
if(powder_flag1.eq.4.and.d32_d43_flag.eq.1) then
   C0=1.67d11 !C0 for Mannitol using monodisperse initial particles
            with size of d32=100.9 um (dimensionless v = 5)
   write (171,*) "C0 =", C0
   !!!v1=5.0
   write (171,*) "v1 =", v1
end if

if(powder_flag1.eq.4.and.d32_d43_flag.eq.2) then
   C0=5.49d10 !C0 for Mannitol–ATAB 50:50 wt% blend
   !!!C0=7.91d10 !C0 for Mannitol using monodisperse initial
            particles with size of d43=129.6 um (dimensionless v = 10)
   write (171,*) "C0 =", C0
   !!!v1=10.0
   write (171,*) "v1 =", v1
end if

if(powder_flag1.eq.2.and.d32_d43_flag.eq.1) then
   C0=7.92d10 !C0 for A–TAB using monodisperse initial particles
            with size of d32=129.5 um (dimensionless v = 10)
```

write (171,*) "C0 =", C0
!!!v1=10.0
write (171,*) "v1 =", v1
end if

if (powder_flag1.eq.2 .and. d32 .and. d43) then
  C0=3.07d10 !C0 for A-TAB using monodisperse initial particles
  with size of d43=177.5 um (dimensionless v = 26)
  write (171,*) "C0 =", C0
  !!!v1=26.0
  write (171,*) "v1 =", v1
end if

if (powder_flag1.eq.1 .and. d32 .and. d43) then
  C0=3.78d10 !C0 for Avicel using monodisperse initial particles
  with size of d32=165.7 um (dimensionless v = 21)
  write (171,*) "C0 =", C0
  !!!v1=21.0
  write (171,*) "v1 =", v1
end if

if (powder_flag1.eq.1 .and. d32 .and. d43) then
  C0=3.30d10 !C0 for Avicel using monodisperse initial particles
  with size of d43=173.4 um (dimensionless v = 24)
  write (171,*) "C0 =", C0
  !!!v1=24.0
  write (171,*) "v1 =", v1
end if

!!C0=2.71d10
!!write (171,*) "C0 =", C0

!print *, 'Enter V_R (reactor volume, m^3) in decimal form (e.g., 0.01)'
!read *, V_R
V_R=1.0d-2
write (171,*) "V_R =", V_R

!print *, 'Enter m_air (mass flowrate of drying air, kg/min) in decimal form (e.g., 2.0)'


!read *, m_air
m_air=2.0
write (171,*) "m_air =", m_air

!print *, 'Enter rho_air (density of drying air, kg/m^3) in decimal form (e.g., 1.184)'
!read *, rho_air
rho_air=1.184d0
write (171,*) "rho_air =", rho_air

!print *, 'Enter Q_in (volumetric flow of air in, m^3/s) in decimal form (e.g., 0.001)'
!read *, Q_in
Q_in=m_air/60.0/rho_air
!Q_in=0.027372d0
!write(*,*) "Q_in =", Q_in
write (171,*) "Q_in =", Q_in

!print *, 'Enter Cn0 (number conc of particles in the binder flow in, 1/m^3 air in) in decimal form (e.g., 1.696d9)'
!read *, Cn0
Cn0=6.013536d7
write (171,*) "Cn0 =", Cn0

!Use the Antoine eqn for water to determine the mass fraction of water in wet inlet air
!print *, 'Enter dewpoint temp of inlet air in degrees Celsius and decimal form (e.g., 25.0)'
!read *, Tdew_C
Tdew_C=10.0d0
!write (171,*) "Dewpoint of inlet air =", Tdew_C
Tdew_K=Tdew_C+273.15d0

Psat_dew = exp(21.8261d0 - (3130.7984d0/(Tdew_K-70.657d0)))

MW_h2o=18.0d0
MW_air=29.0d0
dew_mol_ratio=Psat_dew/(101325.0d0-Psat_dew)
!!inlet_humidity = MW_h2o/MW_air * (dew_mol_ratio/(1+(MW_h2o/MW_air)*dew_mol_ratio))
inlet_humidity =0.0074  !0.006455d0
write (171,*) "inlet_humidity =", inlet_humidity

!print *, 'Enter initial amount of moisture (dimensionless) in the
granulator in decimal form (e.g., 0.0)'
!read *, moisture
moisture=0.0
!!moisture=inlet_humidity
write (171,*) "moisture =", moisture

!print *, 'Enter cw0 (dimensionless mass conc of moisture in the
air of the binder flow in) in decimal form (e.g., 0.001)'
!read *, cw0
cw0=(rho_air*inlet_humidity)/(C0*(rho*(3.14159d0*(d0**3.0d0)/6.0d0))
)
!write (*,*) "cw0 =", cw0
write (171,*) "cw0 =", cw0

!Use the Antoine eqn for water to determine the saturation mass
fraction of water in wet granulator air
!print *, 'Enter temp of granulation bed in degrees Celsius and
decimal form (e.g., 25.0)'
!read *, Tsat_C
Tsat_C=25.0d0  !27.55d0  !25.0d0
write (171,*) "Temp of granulation bed =", Tsat_C
Tsat_K=Tsat_C+273.15d0

Psat_bed = exp(21.8261d0 - (3130.7984d0/(Tsat_K-70.657d0)))
sat_mol_ratio=Psat_bed/(101325.0d0-Psat_bed)
sat_frac= MW_h2o/MW_air * (sat_mol_ratio/(1+(MW_h2o/MW_air)*sat_mol_ratio))
!!sat_frac =0.0195
write (171,*) "sat_frac =", sat_frac

!print *, 'Enter cw_sat (dimensionless saturated mass conc of
moisture in the granulator) in decimal form (e.g., 0.01)'

!read *, cw_sat

\[ cw_{sat} = \frac{(\rho_{\text{air}} \times \text{sat}_{\text{frac}})}{(C0 \times (\rho_{\text{sat}}(3.14159d0 \times (d0**3.0d0)/6.0d0)))} \]

!write(*,*) "cw_sat =", cw_sat

write (171,*) "cw_sat =", cw_sat

!print *, 'Enter N_spray (binder flow rate, #/s) in decimal form (e.g., 1.696d6)

!read *, N_spray

!N_spray=1.696d6
N_spray=Q_in*cn0
write (171,*) "N_spray =", N_spray

!print *, 'Enter Q_out (volumetric flow of air out, m^3/s) in decimal form (e.g., 0.001)

!read *, Q_out
Q_out=Q_in
write (171,*) "Q_out =", Q_out

\[ \tau_{c} = \frac{1}{(K_{PP} \times C0)} \] ! Calculation of characteristic coagulation time (units of s)
write(*,*) "\tau_{c} =", \tau_{c}
write (171,*) "\tau_{c} =", \tau_{c}

\[ \tau_{in} = \frac{V_R}{Q_{in}} \] ! Calculation of characteristic flow in time (units of s)
write(*,*) "\tau_{in} =", \tau_{in}
write (171,*) "\tau_{in} =", \tau_{in}

!print *, 'Enter Tau_coag/Tau_in ratio in decimal form (e.g., 0.1)

!read *, tau_c/tau_in
\[ \tau_{c/tau_in} = \frac{\tau_{c}}{\tau_{in}} \] ! Calculation of dimensionless coagulation time to flow in time ratio
!!\tau_{c/tau_in}=0.0608
write(*,*) "\tau_{c/tau_in} =", \tau_{c/tau_in}
write (171,*) "\tau_{c/tau_in} =", \tau_{c/tau_in}

\[ \tau_{out} = \frac{V_R}{Q_{out}} \] ! Calculation of characteristic flow out time (units of s)
write(*,*) "\tau_{out} =", \tau_{out}
write (171,*) "\tau_{out} =", \tau_{out}
!print *, 'Enter Tau_coag/Tau_out ratio in decimal form (e.g., 0.1),

!read *, tau_c, tau_out ! Calculation of dimensionless
coaulation time to flow out time ratio
write (171,*) "tauc_tauratio =", tau_c/tau_out

eps=nu_binder1/(Cn0/C0) ! Defining arbitrary flow in volume
during MC event (Coarsening factor)
write (171,*) "eps_in =", eps
eps_out=eps_in ! Defining arbitrary flow out volume
during MC event (Coarsening factor)
write (171,*) "eps_out =", eps_out

!write(*,*) "rate in =", tau_c/eps_in
!write(*,*) "rate out =", tau_c/eps_out

evap_area=3.14159d0*(d0**2.0d0)
write (171,*) "evap_area =", evap_area

!print *, 'Enter evap_gamma (adjustable parameter used in
evaporation rate) in decimal form (e.g., 0.02),
!read *, evap_gamma

D_water=(2.19d-5)*((Tsatu/K/273.1d0)**1.75d0) ! Diffusivity of
water vapor in air (m^2/s)
write (171,*) "D_water =", D_water

mass_k=2.0d0*D_water/((v1**((1.0d0/3.0d0)))*d0) ! Initial mass
transfer coefficient (m/s)
write (171,*) "mass_k =", mass_k

tau_evap=1/(evap_area*C0*evap_gamma*mass_k) ! 1.0d0 ! Defining
characteristic evaporation time (units of s)
write(*,*) "tau_evap =", tau_evap

!print *, 'Enter evap_gamma (adjustable parameter used in
evaporation rate) in decimal form (e.g., 0.02),
!read *, evap_gamma

tauc_tauratio=tau_c/tau_out ! Calculation of dimensionless
coaulation time to flow out time ratio
write (171,*), "tauevap =", tauevap

!print *, 'Enter evap_mass(dimensionless mass of water removed from
binder per evaporation event) in decimal form (e.g., 1.0d-1)'
!read *, evap_mass

! Defining arbitrary evaporation mass during MC
! event (Coarsening factor)
write (171,*), "evap_mass =", evap_mass

n=10000 ! total number of sim particles
write (171,*), "n =", n
maxloop=980000 ! Max number of loop events (if simulation is
governed by number of loop events)
print_freq=2000 ! printing frequency
write (171,*), "print_freq =", print_freq

! Write initial values to various input files

! write (01,*), "Number of particles =", n
! write (01,*), "v1 =", v1
! write (01,*), "vb =", vb
! write (01,*), "wb =", wb
! write (01,*), "wp =", wp
! write (01,*), "Number fraction of particles of powder A =", fa
phi0=n*(1.0-fa)*vb/((n*fa*v1)+(n*(1.0-fa)*vb))
write (171,*), "phi0 =", phi0
beta=v1/vb
write (171,*), "beta =", beta
! write (01,*), "Beta (v1/vB) =", beta
! write (01,*), "Average v0 (initial avg mass of system) =", avg_v0
! write (01,*), ""
! write (01,*), " Particle mass Binder mass Water mass"

write (31,*), "Number of particles =", n
write (31,*), "v1 =", v1
write (31,*), "vb =", vb
write (31,*), "wb =", wb
write (31,*), "wp =", wp
write (31,*) "Number fraction of particles of powder A =", fa
write (31,*) "Phi0 (initial mass fraction of solute) =", phi0
write (31,*) "Beta (v1/vB) =", beta
!
write (31,*) "Phi0 (initial mass fraction of solute) =", phi0
!
write (31,*) "Average v0 (initial avg mass of system) =", avg_v0
write (31,*) "Tau ratio (Tau_coag/Tau_in) =", tauc_tauin

write (31,*) "Number of binder droplets added per event =", n_binder1
write (31,*) ""
write (31,*) "Wet gran mass Granule solute conc"
write (141,*) "Number of particles =", n
write (141,*) "v1 =", v1
write (141,*) "vb =", vb
write (141,*) "wb =", wb
write (141,*) "wp =", wp
write (141,*) "Number fraction of particles of powder A =", fa
write (141,*) "Phi0 (initial mass fraction of solute) =", phi0
write (141,*) "Beta (v1/vB) =", beta
!
write (141,*) "Average v0 (initial avg mass of system) =", avg_v0
write (141,*) ""
write (141,*) "Gran size (by vol) Powder 1 mass Powder 2 mass"
write (141,*) "Gran size (by mass) Powder 1 mass frac Binder mass &"
write (141,*) "& Original Moisture mass Moisture mass Dry Gran mass"
write (141,*) "Dried solute conc Wet solute conc HPC mass frac &"
write (141,*) "& Granule Density"
write (81,*) ""
write (81,*) "Tau ratio (Tau_coag/Tau_in) =", tauc_tauin
write (81,*) "Number of binder droplets added per event =", n_binder1
write (81,*) ""
write (81,*) "Vavg (all) # Conc"
write (91,*) "Time Avg eta Avg eta_ap"
write (91,*) "m_evap"
write (91,*) ""
write (91,*) "Tau ratio (Tau_coag/Tau_in) =", tauc_tauin
write (91,*) "Number of binder droplets added per event =", n_binder1
write (91,*) ""
write (91,*) "cB cP c"
write (91,*) ""
write (101,*) ""
write (101,*) "Tau ratio (Tau_coag/Tau_in) =", tauc_taulin
write (101,*) "Number of binder droplets added per event =", n_binder1
write (101,*) ""
write (101,*) " Wet gran mass Granule solute conc Dried gran mass Powder only mass"
write (131,*) ""
write (131,*) "Tau ratio (Tau_coag/Tau_FS) =", tau_ratio
write (131,*) "Number of binder droplets added per event =", n_binder1
write (131,*) ""
write (131,*) " Wet gran mass Granule solute conc Dried gran mass"
write (151,*) "Number of particles =", n
write (151,*) "v1 =", v1
write (151,*) "vb =", vb
write (151,*) "wb =", wb
write (151,*) "wp =", wp
write (151,*) "Number fraction of particles of powder A =", fa
write (151,*) "Phi0 (initial mass fraction of solute) =", phi0
write (151,*) "Beta (v1/vB) =", beta
write (151,*) ""
write (151,*) " Particle number conc Particle mass conc Moisture mass conc Dimensionless time"

***** Initialize settings *****

! initialize particle array (bidisperse conditions)
print *, 'v1 = ', v1
print *, 'vb = ', vb
print *, 'wb = ', wb
print *, 'wp = ', wp
\[ \text{fa} = \frac{\phi_0}{(\beta + \phi_0 - \beta \phi_0)} \]

\[ \text{v} = \text{v}_1 \quad \text{initial particle mass (equals excipient mass)} \]

\[ \text{m} = 0.0 \quad \text{initial all solute to solute} = 0 \]

\[ \text{w} = \text{wp} \quad \text{initial moisture of powder} \]

\[ \text{w}_{m} = \text{wb} \]

\[ \text{do } ii = 1, \text{int}((1.0 - \text{fa})*n) \quad \text{change } \phi_0 \text{ fraction of particles to pure solute} \]

\[ m(ii) = \text{vb} \]

\[ v(ii) = \text{vb} + \text{wb} \]

\[ w(ii) = \text{wb} \]

\[ w_m(ii) = \text{wb} \]

\[ \text{end do} \]

\[ \text{do } ii = 1, \text{int}((1.0 - \text{fa})*n) \quad \text{change } \phi_0 \text{ fraction of particles to pure solute} \]

\[ m(ii) = \text{vb} \]

\[ v(ii) = \text{vb} + \text{wb} \]

\[ w(ii) = \text{wb} \]

\[ w_m(ii) = \text{wb} \]

\[ \text{end do} \]

**Initial setting of particle with largest mass**

\[ \text{if } (\text{v}_1 \gt \text{vb} + \text{wb} \text{. and } \text{v}_1 \gt \text{v}_2) \text{ then} \]

\[ \text{vmax} = \text{v}_1 \]

\[ \text{else if } (\text{v}_2 \gt \text{vb} + \text{wb} \text{. and } \text{v}_2 \gt \text{v}_1) \text{ then} \]

\[ \text{vmax} = \text{v}_2 \]

\[ \text{else if } (\text{vb} + \text{wb} \gt \text{vb} + \text{wb} \text{. and } \text{vb} + \text{wb} \gt \text{v}_1) \text{ then} \]

\[ \text{vmax} = \text{vb} \]

\[ \text{else if } (\text{vb} + \text{wb} \gt \text{vb} + \text{wb} \text{. and } \text{vb} + \text{wb} \gt \text{v}_2) \text{ then} \]

\[ \text{vmax} = \text{vb} \]

\[ \text{end do} \]

**Initial setting of solute with largest mass**

\[ \text{mmax} = \text{vb} + \text{wb} \]

**Initial setting of particle with smallest mass**

\[ \text{if } (\text{v}_1 \lt 1.0) \text{ then} \]

\[ \text{v}_1 = 1.0 \]

\[ \text{end if} \]

\[ \text{if } (\text{v}_1 \lt \text{vb} \text{. and } \text{v}_1 \lt \text{v}_2) \text{ then} \]

\[ \text{vmin} = \text{v}_1 \]

\[ \text{else if } (\text{v}_2 \lt \text{vb} \text{. and } \text{v}_2 \lt \text{v}_1) \text{ then} \]

\[ \text{vmin} = \text{v}_2 \]

\[ \text{else if } (\text{vb} + \text{wb} \lt \text{vb} + \text{wb} \text{. and } \text{vb} + \text{wb} \lt \text{v}_1) \text{ then} \]

\[ \text{vmin} = \text{vb} \]

\[ \text{else if } (\text{vb} + \text{wb} \lt \text{vb} + \text{wb} \text{. and } \text{vb} + \text{wb} \lt \text{v}_2) \text{ then} \]

\[ \text{vmin} = \text{vb} + \text{wb} \]

**Use when evaporation doesn’t change mass of granule (e.g., when comparing with RM)**
else
    vmin=v1
end if
!
write(*,*) "Vmax =", vmax
!
write(*,*) "Vmin =", vmin
!
Initial setting of solute with smallest mass
mmin=0

**** ECHO conditions ****
lola=kernel(1,1) ! call kernel to get kernel_type
write(*,*) "-----------------------------"
write(*,*) "* * * Input Conditions * * *"
write(*,*) "kernel: ", kernel_type
write(*,*) "n=", n, "phi0=", phi0
write(*,*) "maxloop=", maxloop, "print_freq=", print_freq
write(*,*) "-----------------------------"
write(*,*) "Stats of initial distribution"

! Calculate initial average mass
mass=0.0
do ii=1,n
    mass=mass+v_m(ii)
end do
vav=mass/(1.0*n)

! Set mass concentration equal to average mass (since vav = mass
conc / # conc and # conc is 1 initially)
mass_conc=vav
conc=1.0

time=0.0  ! Set time equal to zero
binder=0.0  ! Set counter for the # of binder particles equal to zero
part=1.0*n  ! Set counter for the # of granules equal to zero
loop=0
!!! hh=0  ! Set counter for averaging multiple runs to zero
call stats  ! print stats of initial distribution
write(*,*)
! Record mass data of initial distribution *****
! do ii=1, n
! write (01,*) v(ii), m(ii), w(ii)
! end do

***** PERFORM EVENT *****

! Set dummy variables equal to zero
ii=0
kk=0
mm=0
lll=1

! Set counters for MC events equal to zero
binder_selection=0 ! May be different than flowin_count if # of
! binder droplets introduced per event is < 1
coag_count=0
flowin_count=0
flowout_count=0
evap_count=0

! Establish desired duration of simulation
! do while (vav.le.30.0)
 do while (time.le.3300.0/tau_c)
ni=ii+1
 loop=ii
! write(*,*), "Loop ", ii
! do ii=1, maxloop

! Calculate before event average mass
mass=0.0
 do jj=1,n
   !mass=mass+v(jj)
mass=mass+v_m(jj)
 end do
vav=mass/(1.0*n)
! write(*,*) "mass before event = ", mass ! Used for debugging
purposes
! Determine value of $x_{43}$
! Determine sum of $v_{43}$

$v_{43} = 0.0$

**do** jj=1, n

$v_{43} = v_{43} + v(jj) \times (4.0/3.0)$

**end do**

$x_{43} = v_{43}/\text{mass}$

$! x_{43}=1$

! *write(*,*) "v43 = ", v43 ! Used for debugging purposes

! *write(*,*) "x43 = ", x43 ! Used for debugging purposes

**call** event_selection

**if** (coag.eq.1) **then**

$\text{coag\_count} = \text{coag\_count} + 1$ ! Variable that tracks the # of coagulation events

**call** coagulation

! *write(*,*) "loop ", ii , ", = coagulation"

**elseif** (coag.eq.2) **then**

$\text{flowin\_count} = \text{flowin\_count} + 1$ ! Variable that tracks the # of flow in events

$\text{binder} = \text{binder} + n\_\text{binder1}$ ! Variable that tracks the # of binder droplets in the simulation box

$\text{part} = \text{part} - n\_\text{binder1}$ ! Variable that tracks the # of granules in the simulation box

**call** binder_flow

!! if (binder_flag.eq.1) then

!! $\text{binder} = \text{binder} + n\_\text{binder2}$

!! $\text{part} = \text{part} - n\_\text{binder2}$

!! kk=kk+1

!! end if

!! *write(*,*) "loop ", ii , ", = binder addition"

kk=kk+1

**elseif** (coag.eq.3) **then**

$\text{flowout\_count} = \text{flowout\_count} + 1$ ! Variable that tracks the # of flow out events

**call** Calc_updates

! *write(*,*) "loop ", ii , ", = flow out"

mm=mm+1

**else**
evap_count=evap_count+1  ! Variable that tracks the # of evaporation events
call evaporation
! ! write(*,*) ”loop ”, ii, ”= evaporation”
end if

! Statement to write stats to screen and output files at an interval equal to print_freq
if(mod(ii,print_freq).eq.0) then
! write(*,*) ”loop #”, ii
call stats
! write(*,*) ”moisture =”, moisture
! write(*,*)
end if

kkk=time*(1/(K_PP*C0))

! Statement to write simulation status to screen at regular intervals
if(mod(kkk,100).eq.0) then
  if(lll.eq.1) then
    call date_and_time(date_time_zone(1), date_time_zone(2), date_time_zone(3), date_time)
    write(*,*) ”Time =”, time*(1/(K_PP*C0)), ”s at ”&
    & ,date_time(5),”:”, date_time(6),”:”, date_time(7), ”on ”&
    & ,date_time(2),’/’, date_time(3),’/’, date_time(1)
  end if
end if
lll=lll+1
if(mod(kkk,100).ne.0) then
  lll=1
end if

if(mod(ii,1).eq.0) then
  ! ! write(*,* ) ”Loop ”, loop, ”completed”
end if
end do

write(*,*) ”Stats of final distribution”
call stats
Record number of max kernel violations in log

write (51,*), 'There are ', rr, ' max kernel violations in this run of ', ii, ' loops!!!'

Write statistics to screen and output files

write(*,*) 'Average kernel =', average_kernel
write(*,*) 'binder droplets =', binder
write(*,*) 'Number of binder flow in events =', kk
write(*,*) 'Number of binder droplets in =', binder_selection
write(*,*) 'Moist_in =', n_binder1*cw0/(Cn0/C0)
write(*,*) 'Number of flow out events =', mm
write(*,*) 'Moisture in granulator =', moisture
write(*,*) 'Number of coagulation events =', coag_count
write(*,*) 'Number of flow in events =', flowin_count
write(*,*) 'Number of flow out events =', flowout_count
write(*,*) 'Number of evaporation events =', evap_count
!! write(*,*) 'Total number of events =', loop

""
write(121,*), "Average kernel =", average_kernel
write(121,*), "binder droplets =", binder
write(121,*), "Number of actual binder flow in events =", kk
write(121,*), "Number of coagulation events =", coag_count
write(121,*), "Number of binder events =", flowin_count
write(121,*), "Total number of events =", loop

Record mass and concentration data for final distribution

write (*,155) r

open (3, file = 'Final Dist Sieves.txt', status='old')

open (4, file = FileName14)

! open (4, file = 'Final Distribution.txt') ! For storage of initial distribution
write(4,*), 'FINAL PARTICLE DISTRIBUTION'
write(4,*) ""

! Determine # of sieves to be used for final distribution
sieve_count=0
do i=1,10000
  read (3,*,end=35) (sieves(i))
  !write(*,*), "Sieve ",i,"=" ,sieves(i)
sieve_count=sieve_count+1
end do
Continue

! Sort each granule into the appropriate sieve cut
do j=1, sieve_count
  sieve_number(j)=0
  do k=1,n
    if (j.eq.1) then
      !if (v(k).le.sieves(j)) then  ! For sieve cuts by volume
      if (v(k).le.((sieves(j)**3.0)/(60.0**3.0))) then  ! For sieve cuts by diameter
        sieve_number(j)=sieve_number(j)+1
      end if
    end if
  end if
  if (j.gt.1.and.j.lt.sieve_count) then
    !if (v(k).gt.sieves(j-1).and.v(k).le.sieves(j)) then  ! For sieve cuts by volume
    if (v(k).gt.((sieves(j-1)**3.0)/(60.0**3.0)).and.v(k).le.((sieves(j)**3.0)/(60.0**3.0))) then
      sieve_number(j)=sieve_number(j)+1
    end if
  end if
  if (j.eq.sieve_count) then
    !if (v(k).gt.sieves(j-1).and.v(k).le.sieves(j)) then  ! For sieve cuts by volume
    if (v(k).gt.((sieves(j-1)**3.0)/(60.0**3.0)).and.v(k).le.((sieves(j)**3.0)/(60.0**3.0))) then
      sieve_number(j)=sieve_number(j)+1
    end if
  end if
  if (v(k).gt.sieves(j)) then  ! For sieve cuts by volume
if \((v(k).gt.((\text{sieves}(j)**3.0)/(60.0**3.0)))\) then ! For sieve
cuts by diameter
\[
s\text{i}e\text{v}e\text{ }\text{n}\text{u}m\text{e}\text{r}(j+1)=s\text{i}e\text{v}e\text{ }\text{n}\text{u}m\text{e}\text{r}(j+1)\text{+}1
\]
end if
\[
s\text{i}e\text{v}\text{e}s(j+1)=s\text{i}e\text{v}\text{e}s(j)
\]
end if
\]
end do
\]
end do
\]

! Write the final distribution sieve information to the screen and
to an output file
do \text{i}=1,\text{sieve\_count+1}
! write(\ast,\ast) "\# of particles in sieve ", i, ", ", \text{sieve\_number}(i)
!! write(\ast,\ast) "\# of particles in sieve =", sieves(i), ",\text{"wm ="},
\text{sieve\_number}(i)
write(4,\ast) sieves(i), ", ", \text{sieve\_number}(i)
end do
\]
\]
\]
close(3)
close(4)

! Write array of granules to output files (31 is for all particles,
101 excludes binder-only particles)
! open (31, file = 'output.txt')
do \text{ii}=1, n
!!write (31, '(a,f16.1,a,f16.14,a,f16.1,a)') 
"{", v(ii), ", ", 
1.00d00*m(ii)/v(ii), ", ", part\_gran(ii), ", }",
write (31, '(a,f16.1,a,f16.14,a)') 
"{", v(ii), ", ", 
1.00d00*(m(ii)+w(ii))/v(ii), ", },

!!! if \((1.0d00*m(ii)/v(ii).lt.1.0)\) then
write (101, '(f16.2,a,f16.14,a,f16.2)') v(ii), ", ", 
1.00d00*(m(ii)+w(ii))/v(ii), ", ", v(ii)\-w(ii)
write (131, '(f16.1,a,f16.14,a,f16.2)') v(ii), ", ", 
1.00d00*m(ii)/v(ii), ", 
(v(ii)\-m(ii))+0.15*m(ii)
end if
write (141,\*) v(ii), v_1(ii), v_2(ii), v_m(ii),
powder1\_mass\_frac(ii), m(ii), w(ii), w_m(ii), v(ii)\-w(ii), 
\&
& m(ii)/(v(ii)-w(ii)), (m(ii)+w(ii))/v(ii), HPC_mass_frac(ii),
 rho_gran(ii)
983 ! write(*,*) v(ii), m(ii) ! write particle array on
 screen
end do

! Establish end time of analysis
987 call date_and_time(date_time_zone(1), date_time_zone(2),
 date_time_zone(3), date_time)

989 ! write (01,*) ""
 ! write (01,*) 'Phi0 =', phi0
991 ! write (01,*) 'Ending phi =', phi
 ! write (01,*) 'Percentage of Phi0 ', phi/phi0
993 ! write (01,*) 'aAA =', aAA, 'aAB =', aAB, 'aBB =', aBB
 ! write (01,*) 'End date of analysis (month/day/year): ', date_time(2)
 ,'/', date_time(3), '/', date_time(1)
995 ! write (01,*) 'End time of analysis (hour:minute:second): ',
 date_time(5), ":", date_time(6), ":", date_time(7)

997 ! write (11,*) ""
 ! write (11,*) 'End date of analysis (month/day/year): ', date_time(2)
 ,'/', date_time(3), '/', date_time(1)
999 ! write (11,*) 'End time of analysis (hour:minute:second): ',
 date_time(5), ":", date_time(6), ":", date_time(7)

1001 ! write (21,*) ""
 ! write (21,*) 'End date of analysis (month/day/year): ', date_time(2)
 ,'/', date_time(3), '/', date_time(1)
1003 ! write (21,*) 'End time of analysis (hour:minute:second): ',
 date_time(5), ":", date_time(6), ":", date_time(7)

1005 write (31,*) ""
 write (31,*) 'Phi0 =', phi0
1007 write (31,*) 'Ending phi =', phi
 write (31,*) 'Percentage of Phi0 ', phi/phi0
1009 ! write (31,*) 'aAA =', aAA, 'aAB =', aAB, 'aBB =', aBB
 write (31,*) '# of binder particles =', binder
1011 write (31,*) '# of granules =', part
write (31,*) 'End date of analysis (month/day/year): ', date_time(2),'/',date_time(3),'/',date_time(1)
1013 write (31,*) 'End time of analysis (hour:minute:second): ', date_time(5),':', date_time(6),':', date_time(7)
1015 ! write (41,*) ""
! write (41,*) 'End date of analysis (month/day/year): ', date_time(2),'/',date_time(3),'/',date_time(1)
1017 ! write (41,*) 'End time of analysis (hour:minute:second): ', date_time(5),':', date_time(6),':', date_time(7)
1019 write (141,*) ""
write (141,*) 'Phi0 =', phi0
1021 write (141,*) 'Ending phi =', phi
write (141,*) 'Percentage of Phi0 = phi/phi0
1023 write (141,*) 'aAA =', aAA, 'aAB =', aAB, 'aBB =', aBB
write (141,*) 'End date of analysis (month/day/year): ', date_time(2),'/',date_time(3),'/',date_time(1)
1025 write (141,*) 'End time of analysis (hour:minute:second): ', date_time(5),':', date_time(6),':', date_time(7)
1027 write (81,*) ""
write (81,*) 'End date of analysis (month/day/year): ', date_time(2),'/',date_time(3),'/',date_time(1)
1029 write (81,*) 'End time of analysis (hour:minute:second): ', date_time(5),':', date_time(6),':', date_time(7)
1031 write (91,*) ""
write (91,*) 'End date of analysis (month/day/year): ', date_time(2),'/',date_time(3),'/',date_time(1)
1033 write (91,*) 'End time of analysis (hour:minute:second): ', date_time(5),':', date_time(6),':', date_time(7)
1035 write (101,*) ""
write (101,*) 'End date of analysis (month/day/year): ', date_time(2),'/',date_time(3),'/',date_time(1)
1037 write (101,*) 'End time of analysis (hour:minute:second): ', date_time(5),':', date_time(6),':', date_time(7)
1039 write (111,*) ""
write (111,*), 'End date of analysis (month/day/year): ', date_time(2), '/', date_time(3), '/', date_time(1)
write (111,*), 'End time of analysis (hour:minute:second): ', date_time(5), ':', date_time(6), ':', date_time(7)

write (121,*), ''
write (121,*), 'End date of analysis (month/day/year): ', date_time(2), '/', date_time(3), '/', date_time(1)
write (121,*), 'End time of analysis (hour:minute:second): ', date_time(5), ':', date_time(6), ':', date_time(7)

! write (131,*), ''
! write (131,*), 'End date of analysis (month/day/year): ', date_time(2), '/', date_time(3), '/', date_time(1)
! write (131,*), 'End time of analysis (hour:minute:second): ', date_time(5), ':', date_time(6), ':', date_time(7)

write (151,*), ''
write (151,*), 'End date of analysis (month/day/year): ', date_time(2), '/', date_time(3), '/', date_time(1)
write (151,*), 'End time of analysis (hour:minute:second): ', date_time(5), ':', date_time(6), ':', date_time(7)

write (171,*), ''
write (171,*), 'End date of analysis (month/day/year): ', date_time(2), '/', date_time(3), '/', date_time(1)
write (171,*), 'End time of analysis (hour:minute:second): ', date_time(5), ':', date_time(6), ':', date_time(7)

print *, 'End date of analysis (month/day/year): ', date_time(2), '/', date_time(3), '/', date_time(1)
print *, 'End time of analysis (hour:minute:second): ', date_time(5), ':', date_time(6), ':', date_time(7)

!***** Calculate average values from multiple runs
!!! if (hh.gt.maxdata) then
!!! maxdata=hh
!!! end if
!! do ii=1,maxdata
  ! Gives average dried_d43 across all runs
  run_dried_d43_sum(ii)=(run_dried_d43_sum(ii)+run_dried_d43(ii))
  avg_dried_d43(ii)=run_dried_d43_sum(ii)/r
  ! Gives average time across all runs
  run_time_sum(ii)=(run_time_sum(ii)+run_time(ii))
  avg_time(ii)=run_time_sum(ii)/r
  !!! write(71,*), avg_dried_d43(ii), "", avg_time(ii)
!! end do

!!! write (71,*) ""
!!! write (71,*) 'End date of analysis (month/day/year): ', date_time(2), '/', date_time(3), '/', date_time(1)
!!! write (71,*) 'End time of analysis (hour:minute:second): ', date_time(5), ":", date_time(6), ":", date_time(7)

print *, 'End of simulation ', r

close(31)
close(51)
!!! close(71)
close(81)
close(91)
close(101)
close(111)
close(121)
close(141)
close(151)
close(171)
! close(01)
write(*,*) "All requested runs are now complete."

end program main


A.3 Variable List

```
module particle
  implicit none !
  integer nmax  ! max size of simulation box
  integer sievemax!
  integer eta_number  ! # of random particles to select to calculate
                      ! average eta and eta*ap
  parameter (nmax=50000) !
  parameter (sievemax=10000)!
  parameter (eta_number=100)
  integer n  ! actual size of sim box (set in main)
  integer n_binder2  ! integer number of binder particles introduced
                      ! per event
  integer rr  ! max kernel violation counter variable
  integer coag  ! flag for event selection
  integer loop
  integer binder_selection
  integer binder_flag
  integer powder_flag1  ! flag for establishing A-TAB, Avicel,
                        ! Lactose, or Mannitol as powder 1
  integer powder_flag2  ! flag for establishing A-TAB, Avicel,
                        ! Lactose, or Mannitol as powder 2
  integer d32_d43_flag  ! flag for establishing if monodisperse
                        ! initial size is based on d32 or d43
  integer d43_gamma_flag  ! flag to decide if d43^gamma term is to be
                          ! used
  integer powder_number_flag  ! flag for number of powders (1 or 2)
                              ! to study
  integer hh
  integer eta_holder(eta_number)
  integer eta_counter
  integer moist_count
  real*8 granule_size1 (sievemax)
  real*8 granule_number1 (sievemax)
  real*8 normal1 (sievemax,2)
  real*8 granule_size2 (sievemax)
  real*8 granule_number2 (sievemax)
  real*8 normal2 (sievemax,2)
  real*8 sieves (sievemax)
```
integer sieve_number (sievemax)

!real*8 granule_size (2)  ! For monodisperse distribution
!real*8 granule_number (2)  ! For monodisperse distribution
!real*8 normal (2,2)  ! For monodisperse distribution
!integer granule_size (40)  ! For Gaussian distribution
!integer granule_number (40)  ! For Gaussian distribution
!integer normal (40,2)  ! For Gaussian distribution
!integer granule_size (100)  ! For Log-normal distribution
!integer granule_number (100)  ! For Log-normal distribution
!integer normal (100,2)  ! For Log-normal distribution
!real*8 v(9992)  ! array of particle masses
!real*8 m(9992)  ! array of solute masses
!real*8 v(10004)  ! array of particle masses
!real*8 m(10004)  ! array of solute masses

real*8 n_binder1  ! number of binder particles to be introduced per event

real*8 v(nmax)  ! array of total particle size (by volume)
real*8 v_m(nmax)  ! array of total particle size (by mass)
real*8 v_1(nmax)  ! array of powder 1 particle masses
real*8 v_2(nmax)  ! array of powder 2 particle masses
real*8 powder1_mass_fractions(nmax)  ! array of powder 1 mass fractions
real*8 HPC_mass_fractions(nmax)  ! array of HPC mass fractions
real*8 m(nmax)  ! array of solute masses
real*8 rho_gran(nmax)  ! array of granule densities
real*8 va  ! mass of particles A (solute)
real*8 v1  ! mass of 1st powder particle
real*8 v2  ! mass of 2nd powder particle
real*8 vb  ! mass of particles B (excipient)
real*8 mass  ! total particle mass
real*8 solute  ! total solute mass
real*8 fa  ! fraction of particles A
real*8 vmax  ! particles with largest mass
real*8 vmin  ! particle with smallest mass
real*8 mmax  ! particle with largest solute
real*8 mmin  ! particle with smallest solute
real*8 phi0  ! initial mass fraction of solute
real*8 phi  ! actual mass fraction of solute in sim box
real*8 vav  ! average mass
real*8 time
real*8 deltat
real*8 average_kernel ! average kernel for each loop
real*8 average_kappa    ! average kappa for each loop
real*8 d0    ! mean particle diameter at time zero
real*8 Psi    ! Collision success factor
real*8 gamma    ! x43 exponent
real*8 beta_avicel ! Beta for avicel
real*8 beta_mannitol ! Beta for mannitol
real*8 beta_ATAB    ! Beta for A-TAB
real*8 aAA    ! Activity success factor term
real*8 aAB    ! Activity success factor term
real*8 aBB    ! Activity success factor term
real*8 conc    ! Dimensionless number conc. of particles
real*8 tauc_tuin ! Ratio of Tau_coag/Tau_in (coag over binder
flow in stream)
real*8 tauc_tauout ! Ratio of Tau_coag/Tau_out (coag over flow
out stream)
real*8 tauc_tauevap ! Ratio of Tau_coag/Tau_evap (coag over
evaporation)
real*8 P_coag    ! Probability for selection of coagulation event
real*8 P_flowin ! Probability for selection of flow in event
real*8 P_flowout ! Probability for selection of flow out event
real*8 P_evap    ! Probability for selection of evaporation event
real*8 mass_conc ! Dimensionless mass conc. of particles
real*8 binder
real*8 part
real*8 cB
real*8 cP
real*8 cBcP
real*8 C0    ! Initial concentration of particles in the
            ! granulator
real*8 Cn0    ! Number concentration of particles in the binder
            ! flow in
real*8 cw0    ! Dimensionless mass concentration of moisture in
            ! air of binder flow in stream
real*8 wb    ! Water mass of particles B (excipient)
real*8 initial_LOD1 ! Initial LOD of powder 1
real*8 initial_LOD2 ! Initial LOD of powder 2
real*8 wp    ! Initial water mass of particles A
real*8 w(nmax)    ! Array of moisture volume (assuming
    ! evaporation doesn’t cause volume change)
real*8 w_m(nmax)  ! array of moisture masses
real*8 epsilon_in  ! dimensionless volume of flow into the
    simulation box
real*8 moisture  ! dimensionless amount of moisture in the
    simulation box
real*8 epsilon_out  ! dimensionless volume of flow out of the
    simulation box
real*8 evap_mass  ! dimensionless mass of water evaporated from
    binder per event
real*8 cw_sat  ! dimensionless saturated moisture mass
    concentration in granulator
real*8 binder_water  ! Variable for tracking amount of water in
    particles in granulator
real*8 K_PP
real*8 K_PB
real*8 x43  ! x43 parameter for KTGF kernel
real*8 part_gran (nmax)  ! # of primary particles in a granule
real*8 VR  ! reactor volume
real*8 evap_area  ! characteristic area used in calculation of
    tau_evap
real*8 evap_gamma  ! adjustable parameter used in evaporation rate
real*8 temp  ! Granulation temperature in Kelvin
real*8 D_water  ! Diffusivity of water vapor in air (m^2/s)
real*8 mass_k  ! Initial mass transfer coefficient (m/s)
real*8 m_air  ! Mass flow rate of drying air
real*8 rho_air  ! density of drying air (kg/m^3)
real*8 sat_frac  ! saturation mass fraction of water in air
real*8 inlet_humidity  ! mass fraction of inlet air humidity
real*8 eta_sum  ! used to calculate eta_sum
real*8 average_eta  ! average surface coverage of binder on
    granule
real*8 eta_ap_sum  ! used to calculate average eta_ap
real*8 average_eta_ap  ! average eta_ap of system
real*8 LOD_dry
real*8 ap_sum
real*8 average_ap
real*8 drymass  ! variable for calc of dry granule only vavg
real*8 moisturemass
Variables used in the calculation of the geometric and physical success factors

real*8 bs_ratio1 ! binder/solid ratio for granule 1
real*8 bs_ratio2 ! binder/solid ratio for granule 2
real*8 frc1
real*8 frc2
real*8 eta1 ! accessible binder fraction for granule 1
real*8 eta2 ! accessible binder fraction for granule 2
real*8 a ! empirical constant used in calculation of eta
real*8 a_gran1, a_gran2, a1, a2
real*8 b, b_1, b_2 ! empirical constant(s) used in calculation of eta
real*8 b_gran1, b_gran2, b1, b2
real*8 c, c_1, c_2 ! empirical constant(s) used in calculation of eta
real*8 c_gran1, c_gran2, c1, c2
real*8 geom_Psi ! Geometrical collision success factor
real*8 St ! Stokes number
real*8 rho ! particle density (for calculation of Stokes #)
real*8 mu_cp ! binder viscosity in cp (for calculation of Stokes #)
real*8 mu ! binder viscosity (for calculation of Stokes #)
real*8 u ! velocity of particle collision (for Stokes #)
real*8 critical_St ! Critical Stokes number for evaluation of physical Psi
real*8 lambda1 ! binder layer thickness for granule 1 (for critical Stokes #)
real*8 lambda2 ! binder layer thickness for granule 2 (for critical Stokes #)
real*8 lambda ! Sum of binder layer thicknesses
real*8 phi13_1 ! relative displaced binder volume for granule 1
real*8 phi13_2 ! relative displaced binder volume for granule 2
real*8 f ! empirical constant used in calculation of phi13
real*8 f_gran1, f_gran2, f1, f2
real*8 g ! empirical constant used in calculation of phi13
real*8 g_gran1, g_gran2, g1, g2
real*8 rg ! mean gyration radius
real*8 ha ! surface asperity (for critical Stokes #)
real*8 ha_gran1, ha_gran2, ha1, ha2
real*8 phys_Psi  ! Physical collision success factor
real*8 eta      ! Accessible binder fraction on a particle
real*8 fresh_Psi

end module particle

Appendix–A/particle_module.f95
A.4 Initial Distribution Subroutine

```fortran
subroutine initial_dist ! subroutine that gives a monodisperse, normal, or log-normal initial distribution of powder

!***** Define global variables *****
use particle

!***** Define local variables *****
implicit none

integer i,j,k,ii,jj
integer n1, n2
integer number_of_classes1
integer number_of_classes2

!***** Open necessary input and output files *****
open (1, file = 'Initial Dist Input1.txt', status='old')
open (5, file = 'Initial Dist Input2.txt', status='old')
!
!open (1, file = 'Normal Dist (Avg=20, sigma=5).txt', status='old')
!open (1, file = 'Log-normal Dist (Avg=20, sigma=0.5).txt', status='old')
open (2, file = 'initial distribution.txt') ! For storage of initial distribution

!***** Read particle-particle and particle-binder agglomeration constants, respectively *****
read(1,*) K_PP
!! write(*,*) "K_part_part =", K_PP
read(1,*) K_PB
!! write(*,*) "K_part_bind =", K_PB

!***** Read the initial sieve information for powder 1 *****
n1=0
number_of_classes1=0
do i=1,10000
   read (1,*,end=25) (normal1(i,j), j=1,2)
   granule_size1(i)=normal1(i,1)
   granule_number1(i)=normal1(i,2)
end do
```

---

1. `subroutine initial_dist`: A subroutine that gives a monodisperse, normal, or log-normal initial distribution of powder.
2. `define global variables`: Use of the `particle` module.
3. `define local variables`: Declaration of local variables including integers for indices and counts.
4. `open necessary input and output files`: Opening input files for monodisperse, normal, and log-normal distributions,
   - `open (1, file = 'Initial Dist Input1.txt', status='old')`
   - `open (5, file = 'Initial Dist Input2.txt', status='old')`
   - Additional open statements for normal and log-normal distributions.
5. `read particle-particle and particle-binder agglomeration constants`: Reading constants `K_PP` and `K_PB`.
6. `read the initial sieve information for powder 1`: Loop through the sieve data for powder 1, reading and storing size and number information.
n1 = n1 + granule_number1(i)
number_of_classes1 = number_of_classes1 + 1
end do

Continue
!
!! write(*,*) "Number =", n
!! write(*,*) "# of classes =", number_of_classes

n = 10000 ! For Monodisperse distribution
v = granule_size1 ! For Monodisperse distribution
n = 4996*2 ! For Gaussian distribution
n = 5002*2 ! For Log−normal distribution
!! fa = 1.0 ! percentage of A (powder) particles in decimal form
!! write(171,* ) "fa =", fa

!!! Read the initial sieve information for powder 2 !!!
if (powder_number_flag.eq.2) then
n2 = 0
number_of_classes2 = 0
do i = 1, 10000
read (5,*,end=45) (normal2(i,j), j = 1, 2)
granule_size2(i) = normal2(i,1)
granule_number2(i) = normal2(i,2)
n2 = n2 + granule_number2(i)
number_of_classes2 = number_of_classes2 + 1
end do
Continue
end if

n = n1 + n2
m = 0.0 ! initial mass of binder in each granule
w = wp ! initial moisture of powder
w_m = wp

!!! Form the initial distribution of powder 1 !!!
ii = 1
do j = 1, number_of_classes1
!do j = 1, 2 ! For monodisperse distribution
!do j = 1, 40 ! For Gaussian distribution
!do j = 1, 100 ! For Log−normal distribution
if (granule_number1(j).ne.0) then
do k = 1, granule_number1(j)
  v_1(ii)=granule_size1(j)
  !!!w(ii)=v_1(ii)*initial_LOD1/(1.0d0+initial_LOD1)
  !!!w_m(ii)=w(ii)
  !!!v_1(ii)=v_1(ii)−w(ii)
  ii=ii+1
end do
end if
end do

!***** Form the initial distribution of powder 2 *****
if (powder_number_flag.eq.2) then
  do j = 1, number_of_classes2
    !do j = 1,2 ! For monodisperse distribution
    !do j = 1,40 ! For Gaussian distribution
    !do j = 1,100 ! For Log-normal distribution
    if (granule_number2(j).ne.0) then
      do k = 1, granule_number2(j)
        v_2(ii)=granule_size2(j)
        !!!w(ii)=v_2(ii)*initial_LOD2/(1.0d0+initial_LOD2)
        !!!w_m(ii)=w(ii)
        !!!v_2(ii)=v_2(ii)−w(ii)
        ii=ii+1
      end do
    end if
  end do
end if

v1=maxval(granule_size1)
v2=maxval(granule_size2)
do i=1,n
  v(i)=v_1(i)+v_2(i)
  v_m(i)=v(i)
rho_gran(i)=v_m(i)/v(i)
powder1_mass_frac(i)=v_1(i)/(v_1(i)+v_2(i))
  HPC_mass_frac=0.0d0
end do

!***** Write initial distribution to output file *****


write(2,*) "INITIAL PARTICLE DISTRIBUTION"
write(2,*) "Number of particles =", n
write(2,*) "v1 =", v1
write(2,*) "v2 =", v2
write(2,*) "vb =", vb
write(2,*) "wb =", wb
write(2,*) "wp =", wp
!!write(2,*) "Initial LOD of powder 1 =", initial_LOD1
!!write(2,*) "Initial LOD of powder 2 =", initial_LOD2
write(2,*) "Number fraction of particles of powder A =", fa
phi0=n*(1.0-fa)*vb/((n*fa*va)+(n*(1.0-fa)*vb))
write(2,*) "phi0 =", phi0
write(2,*) "Total Particle mass Powder 1 mass Powder 2 mass Binder mass Water mass"

****** Record mass data of initial distribution ******
do ii=1, n
write(2,*) v(ii), v_1(ii), v_2(ii), m(ii), w(ii)
end do

****** If desired, calculate the initial binder concentration in the simulation box ******
mass=0.0
solute=0.0
do i=1,n
!write(*,*) 'v =', v(i)
!write(*,*) 'm =', m(i)
mass=mass+v(i)
solute=solute+m(i)
part_gran(i)=1.0
!write (2,'(f16.1,a,f16.1)') 1.00d00*v(i), "", "", 1.00d00*m(i)/v(i)
!write (2,'(f16.1,a,f16.1)') 1.00d00*v(i), "", "", 1.00d00*m(i)
!write (2,'(f16.1,a,f16.1)') v(i), "", "", m(i)
end do

!! phi0=solute/mass ! initial mass fraction of binder in decimal form
!! write(*,*) 'phi0 = ', phi0
write(2,*) '
!! write(2,*) 'phi0 = ', phi0
close (1)
close (2)
close (5)
return
end subroutine initial_dist

Appendix–A/Initial_Distribution.f95
A.5 Event Selection Subroutine

```
subroutine event_select

!***** Define global variables *****
use particle

!***** Define local variables *****
implicit none

common/r/iseed
integer trial
integer i, j
real ran0
integer iseed
integer ii, jj
real*8 kernelsum, kernel, kernel2
real*8 denominator, denom1, denom2
real*8 event_selector

!***** Define initial values of key variables *****
coag=1 ! flag for event selection
kernelsum=0.0 ! initialize sum of kernel to be zero (used to
evaluate avg kernel)
binder_water=0.0
write (*,*) "Dimensionless number concentration =", conc ! For
debugging purposes

!***** Evaluate average kernel *****
do ii=1, 1000
  ! pick pair
    trial=ii
    i = 1+n*ran0(iseed)
    if(i.eq.n+1) then ! Needed to prevent a v(10,001) scenario
      i=n
    end if
    j=i
  while (j.eq.i)
    j = 1+n*ran0(iseed)
```

if (j.eq.n+1) then  ! Needed to prevent a v(10,001) scenario
  j=i
end if
end do

! write (*,*) "trial =", trial  ! For debugging purposes
! write (*,*) "v(i) =", v(i)  ! For debugging purposes
! write (*,*) "v(j) =", v(j)  ! For debugging purposes
! write (*,*) "kernel(i,j) =", kernel(i,j)  ! For debugging purposes

!!kernelsum = kernelsum+kernel(i,j)
! write (*,*) "kernelsum =", kernelsum  ! For debugging purposes

!!bs_ratio1 = m(i)/(v(i)-m(i))
!!bs_ratio2 = m(j)/(v(j)-m(j))

**** Calculate simple sigmoidal-correlation Psi factor ****
!! if (v(i).eq.(m(i)+w(i)).and.v(i).ne.m(i)) then
!!  bs_ratio1 = 1.0
!! else
!!  bs_ratio1 = w(i)/(v(i)-w(i))  ! Use if ratio is to be eq. binder content over dry granule content
!!  !!!!bs_ratio1 = (m(i)+w(i))/(v(i)-(m(i)+w(i))) ! Use if ratio is to be entire binder content over powder content
!! end if

!! if (v(j).eq.(m(j)+w(j)).and.v(j).ne.m(j)) then
!!  bs_ratio2 = 1.0
!! else
!!  bs_ratio2 = w(j)/(v(j)-w(j))  ! Use if ratio is to be eq. binder content over dry granule content
!!  !!!!bs_ratio2 = (m(j)+w(j))/(v(j)-(m(j)+w(j))) ! Use if ratio is to be entire binder content over powder content
!! end if

!! if (bs_ratio1.gt.0.735) then
!!  frc1=1.0
!! end if

!! if (bs_ratio1.le.0.735) then
!!  frc1=1.0291*((1.0-exp(-6.8539*bs_ratio1))**4.4089)
!! end if

!! if (bs_ratio1.eq.0.0) then
!! frc1=0.0
!! end if

!! if (bs_ratio2.gt.0.735) then
!! frc2=1.0
!! end if

!! if (bs_ratio2.le.0.735) then
!! frc2=1.0291*((1.0-exp(-6.8539*bs_ratio2))**4.4089)
!! end if

!! if (bs_ratio2.eq.0.0) then
!! frc2=0.0
!! end if

!! Psi=1.0-(1.0-frc1)*(1.0-frc2)

!**** Calculation of geometrical Psi ****
!**** Calculation of etas ****

!**** "a" values ****
if(powder_flag1.eq.1) then
  a1=1.0    ! Avicel value
elseif(powder_flag1.eq.2) then
  a1=1.0    ! A-TAB value
elseif(powder_flag1.eq.3) then
  a1=1.0    ! Lactose value
elseif(powder_flag1.eq.4) then
  a1=1.0    ! Mannitol value
end if

if(powder_flag2.eq.1) then
  a2=1.0    ! Avicel value
elseif(powder_flag2.eq.2) then
  a2=1.0    ! A-TAB value
elseif(powder_flag2.eq.3) then
112 \[ a_2 = 1.0 \quad ! \text{Lactose value} \]
113 `else if (powder_flag2.eq.4) then`
114 \[ a_2 = 1.0 \quad ! \text{Mannitol value} \]
115 `end if`
117
118 \[ a_{\text{gran}1} = \text{powder1_mass_frac}(i) \times a_1 + (1.0d0 - \text{powder1_mass_frac}(i)) \times a_2 \]
119 \[ a_{\text{gran}2} = \text{powder1_mass_frac}(j) \times a_1 + (1.0d0 - \text{powder1_mass_frac}(j)) \times a_2 \]
120
121 \*[**** "b" values ****]
122 `if (powder_flag1.eq.1) then`
123 \[ b_1 = 35.1 \quad ! \text{Avicel value} \]
124 `elseif (powder_flag1.eq.2) then`
125 \[ b_1 = 33.5 \quad ! \text{A-TAB value} \]
126 `elseif (powder_flag1.eq.3) then`
127 \[ b_1 = 33.0 \quad ! \text{Lactose value} \]
128 `elseif (powder_flag1.eq.4) then`
129 \[ b_1 = 27.4 \quad ! \text{Mannitol value} \]
130 `end if`
131
132 `if (powder_flag2.eq.1) then`
133 \[ b_2 = 35.1 \quad ! \text{Avicel value} \]
134 `elseif (powder_flag2.eq.2) then`
135 \[ b_2 = 33.5 \quad ! \text{A-TAB value} \]
136 `elseif (powder_flag2.eq.3) then`
137 \[ b_2 = 33.0 \quad ! \text{Lactose value} \]
138 `elseif (powder_flag2.eq.4) then`
139 \[ b_2 = 27.4 \quad ! \text{Mannitol value} \]
140 `end if`
141
142 \[ b_{\text{gran}1} = \text{powder1_mass_frac}(i) \times b_1 + (1.0d0 - \text{powder1_mass_frac}(i)) \times b_2 \]
143 \[ b_{\text{gran}2} = \text{powder1_mass_frac}(j) \times b_1 + (1.0d0 - \text{powder1_mass_frac}(j)) \times b_2 \]
144
145 \*[**** "c" values ****]
146 `if (powder_flag1.eq.1) then`
147 \[ c_1 = 0.129 \quad ! \text{Avicel value} \]
148 `elseif (powder_flag1.eq.2) then`
149 \[ c_1 = 0.333 \quad ! \text{A-TAB value} \]
150 `elseif (powder_flag1.eq.3) then`
151 \[ c_1 = 0.206 \quad ! \text{Lactose value} \]
152 `elseif (powder_flag1.eq.4) then`
c1=0.182 ! Mannitol value

if(powder_flag2.eq.1) then
  c2=0.129 ! Avicel value
elseif(powder_flag2.eq.2) then
  c2=0.333 ! A-TAB value
elseif(powder_flag2.eq.3) then
  c2=0.206 ! Lactose value
elseif(powder_flag2.eq.4) then
  c2=0.182 ! Mannitol value
end if

c_gran1=powder1_mass_frac(i)*c1+(1.0d0-powder1_mass_frac(i))*c2

c_gran2=powder1_mass_frac(j)*c1+(1.0d0-powder1_mass_frac(j))*c2

if (v(i).eq.(m(i)+w(i)).and.v(i).ne.m(i)) then
  bs_ratio1 = 1.0
else
  bs_ratio1 = w(i)/(v(i)-w(i)) ! Use if ratio is to be aq. binder content over dry granule content
  bs_ratio1 = (m(i)+w(i))/(v(i)-(m(i)+w(i))) ! Use if ratio is to be entire binder content over powder content
end if

if (v(j).eq.(m(j)+w(j)).and.v(j).ne.m(j)) then
  bs_ratio2 = 1.0
else
  bs_ratio2 = w(j)/(v(j)-w(j)) ! Use if ratio is to be aq. binder content over dry granule content
  bs_ratio2 = (m(j)+w(j))/(v(j)-(m(j)+w(j))) ! Use if ratio is to be entire binder content over powder content
end if

eta1=a_gran1/(1.0+EXP(-b_gran1*(bs_ratio1-c_gran1)))
eta2=a_gran2/(1.0+EXP(-b_gran2*(bs_ratio2-c_gran2)))

! write(*,* )"eta1 = ", eta1 ! used for debugging purposes
! write(*,* )"eta2 = ", eta2 ! used for debugging purposes
178

geom_Psi = 1 - (1-eta1)*(1-eta2)

190

!***** Calculation of physical Psi *****

192

!!St = (4.0/9.0)*(rho*u/mu)*d0*(((v(i)**(1.0/3.0))*(v(j)**(1.0/3.0))+(v(i)**(1.0/3.0))*(v(j)**(1.0/3.0))))**2.0)/(v(i)+v(j))

St = (4.0/9.0)*(u/mu)*rho_gran(i)*rho_gran(j)*rho)*d0*(((v(i)**(1.0/3.0))*(v(j)**(1.0/3.0))))&

&*(((v(i)**(1.0/3.0))+(v(j)**(1.0/3.0))))**2.0)/(rho_gran(i)*rho*v(i)+rho_gran(j)*rho*v(j))

! write(*,*) "v(i) = ", v(i)
196
! write(*,*) "v(j) =", v(j)
! write(*,*) "St =", St

198

!***** Calculation of critical Stokes number *****

!***** Calculation of lambda *****

!***** Calculation of relative displaced binder volume, phi13 *****

202

!***** "f" values *****

! if(powder_flag1.eq.1) then
204
f1=25.9 ! Avicel value
206
elseif(powder_flag1.eq.2) then
208
f1=30.4 ! A-TAB value
210
elseif(powder_flag1.eq.3) then
212
f1=29.8 ! Lactose value
214
elseif(powder_flag1.eq.4) then
216
f1=26.8 ! Mannitol value
218
end if

220

if(powder_flag2.eq.1) then
222
f2=25.9 ! Avicel value
224
elseif(powder_flag2.eq.2) then
226
f2=30.4 ! A-TAB value
228
elseif(powder_flag2.eq.3) then
230
f2=29.8 ! Lactose value
232
elseif(powder_flag2.eq.4) then
234
f2=26.8 ! Mannitol value
236
end if
f_gran1=powder1_mass_frac(i)*f1+(1.0d0−powder1_mass_frac(i))\*f2
f_gran2=powder1_mass_frac(j)*f1+(1.0d0−powder1_mass_frac(j))\*f2

!**** "g" values ****
if(powder_flag1.eq.1) then
  g1=0.327   ! Avicel value
elseif(powder_flag1.eq.2) then
  g1=0.538   ! A-TAB value
elseif(powder_flag1.eq.3) then
  g1=0.403   ! Lactose value
elseif(powder_flag1.eq.4) then
  g1=0.394   ! Mannitol value
end if

if(powder_flag2.eq.1) then
  g2=0.327   ! Avicel value
elseif(powder_flag2.eq.2) then
  g2=0.538   ! A-TAB value
elseif(powder_flag2.eq.3) then
  g2=0.403   ! Lactose value
elseif(powder_flag2.eq.4) then
  g2=0.394   ! Mannitol value
end if

g_gran1=powder1_mass_frac(i)*g1+(1.0d0−powder1_mass_frac(i))\*g2
g_gran2=powder1_mass_frac(j)*g1+(1.0d0−powder1_mass_frac(j))\*g2

phi13_1=EXP(f_gran1*(bs_ratio1−g_gran1))
phi13_2=EXP(f_gran2*(bs_ratio2−g_gran2))

lambda1=rg*(phi13_1**(1.0/3.0))
lambda2=rg*(phi13_2**(1.0/3.0))
lambda=lambda1+lambda2
! write(*,*) "lambda = ", lambda ! used for debugging purposes
!***** Values for ha (input data in micrometers) *****

if (powder_flag1.eq.1) then
  ha1 = 6.3 ! Avicel value
elseif (powder_flag1.eq.2) then
  ha1 = 16.5 ! A-TAB value
elseif (powder_flag1.eq.3) then
  ha1 = 10.5 ! Lactose value
elseif (powder_flag1.eq.4) then
  ha1 = 10.5 ! Mannitol value
end if

if (powder_flag2.eq.1) then
  ha2 = 6.3 ! Avicel value
elseif (powder_flag2.eq.2) then
  ha2 = 16.5 ! A-TAB value
elseif (powder_flag2.eq.3) then
  ha2 = 10.5 ! Lactose value
elseif (powder_flag2.eq.4) then
  ha2 = 10.5 ! Mannitol value
end if

ha_gran1 = powder1_mass_frac(i) * ha1 + (1.0d0 - powder1_mass_frac(i)) * ha2
ha_gran2 = powder1_mass_frac(j) * ha1 + (1.0d0 - powder1_mass_frac(j)) * ha2

ha = (ha_gran1 + ha_gran2) / 2.0d0

critical_St = 2.0 * log(lambda/ha)
!
write(*,*) "Critical Stokes # =", critical_St

if (St.le.critical_St) then
  phys_Psi = 1.0
else
  phys_Psi = 0.0
end if
!
write(*,*) "Physical Psi =", phys_Psi
!
! Warning statement (used for debugging purposes)
if (phys_Psi.eq.0) then
    write(*,*) "!!!!!!!!!!!!!!!!! Physical Psi = 0
!!!!!!!!!!!!!!!!!!!!!!"
end if

*** Calculation of freshness Psi ***
if (HPC_mass_frac(i).ge.0.75 .and. HPC_mass_frac(j).ge.0.75) then
    fresh_Psi=0.0
else
    fresh_Psi=1.0
end if
!!fresh_Psi=1.0 ! Used when you do not wish to consider a freshness Psi factor

Psi=geom_Psi*phys_Psi*fresh_Psi

if ((m(i)+w(i))/v(i).eq.1.0 .and. (m(j)+w(j))/v(j).eq.1.0) then
    kernelsum=kernelsum
elseif ((m(i)+w(i))/v(i).ne.1.0 .and. (m(j)+w(j))/v(j).ne.1.0) then
    kernelsum = kernelsum + kernel(i,j)
elseif ((m(i)+w(i))/v(i).eq.1.0 .and. (m(j)+w(j))/v(j).ne.1.0) then
    kernelsum = kernelsum + kernel2(i,j)
elseif ((m(i)+w(i))/v(i).ne.1.0 .and. (m(j)+w(j))/v(j).eq.1.0) then
    kernelsum = kernelsum + kernel2(i,j)
end if
! write (*,*) "kernelsum =", kernelsum ! For debugging purposes
end do

average_kernel=kernelsum/trial
!write (*,*) "average kernel =", average_kernel ! For debugging purposes

*** For calculation of evaporation rate ***
eta_counter=1

eta_holder(1) = 1+n*ran0(iseed)
if (eta_holder(1).eq.n+1.or.v(eta_holder(1)).eq.1.0d0) then !
    Needed to prevent a v(10,001) scenario
    eta_holder(1)=n

end if

do ii = 2, eta_number
  eta_holder(ii)=eta_holder(1)
  do jj = 1, eta_counter
    if (eta_holder(ii).eq.eta_holder(jj)) then
      eta_holder(ii) = 1+n*ran0(iseed)
      if(eta_holder(ii).eq.n+1.or.v(eta_holder(ii)).eq.1.0d0) then
        ! Needed to prevent a v(10,001) scenario
        eta_holder(ii)=n
      end if
    !write(*,*),"eta_holder",(ii),"=", eta_holder(ii) ! For debugging purposes
    !write(*,*),"eta_holder",(jj),"=", eta_holder(jj) ! For debugging purposes
    go to 125
  end if
  end do
end if  
end do

!if(ii.eq.eta_number) then ! For debugging purposes
!do jj=1,eta_number ! For debugging purposes
!write(*,*),"eta_holder ", jj ," = ", eta_holder(jj) ! For debugging purposes
!end do ! For debugging purposes
!end if ! For debugging purposes
eta_counter = eta_counter +1
end do

ap_sum=0.0
drymass=0.0
moisturemass=0.0

do ii=1,eta_number
  ap_sum=ap_sum+(v(eta_holder(ii))**(2.0/3.0))
  drymass=drymass+v(eta_holder(ii))-w(eta_holder(ii))
  moisturemass=moisturemass+w*m(eta_holder(ii))
end do

average_ap=ap_sum/(1.0d0*eta_number)
LOD_dry=moisturemass/drymass
denom1 = (average_kernel / 2.0) + (tauc_tauin / epsilon_in / (conc**2.0))

if (moisture .ge. (1.0d0 - vb) * cw_sat) then
    moisture = (1.0d0 - vb) * cw_sat
end if

denom2 = (tauc_tauout / epsilon_out / (conc**2.0)) + (tauc_tauvap * LOD_dry * average_ap * ((1.0d0 - vb) * cw_sat - moisture) / evap_mass / conc)

Rajniak new empirical evaporation rate

moist_count = 0

do ii = 1, n
    if (w_m(ii) .ge. evap_mass) then
        moist_count = moist_count + 1
    end if
    binder_water = binder_water + w_m(ii)
end do

if (mod(loop, 1000) .eq. 0) then
    write(*,*) "moist_count =", moist_count Used for debugging
end if

if (moist_count .eq. 0) then
    denom2 = (tauc_tauout / epsilon_out / (conc**2.0)) ! Equation that forces the probability of evaporation to zero
end if

denominator = denom1 + denom2

event_selector = ran0(iseed)

write(*,*) "event selector =", event_selector

P_coag = (average_kernel / 2.0) / denominator

write(*,*) "Probability of coagulation =", P_coag ! For debugging purposes

if (event_selector .le. P_coag) then
    coag = 1
end if

P_flowin = (tauc_tauin / epsilon_in / (conc**2.0)) / denominator

write(*,*) "P_flowin = 0.0"
!! write (*,*) "Probability of flow in =", P_flowin ! For debugging purposes
404 if (event_selector.gt.P_coag.and.event_selector.le.P_coag+P_flowin)
    then
      coag=2
    end if
406
408 !***** FLOW OUT *****
P_flowout=(tauc_tauout/epsilon_out/(conc**2.0))/denominator
410 !P_flowout=0.0
!! write (*,*) "Probability of flow out =", P_flowout ! For debugging purposes
412 if (event_selector.gt.P_coag+P_flowin.and.event_selector.le.P_coag+
P_flowin+P_flowout) then
    coag=3
end if
414
416 !***** EVAPORATION *****
P_evap=(tauc_taufeav*LOD_dry*average_ap*((1.0d0-vb)*cw_sat -
    moisture)/evap_mass/conc)/denominator
418 !P_evap=1.0
if (moist_count.eq.0) then
    P_evap=0.0
end if
420 ! write (*,*) "binder_water =", binder_water ! For debugging purposes
422 ! write (*,*) "Probability of evaporation =", P_evap ! For debugging purposes
424 if (event_selector.gt.P_coag+P_flowin+P_flowout.and.event_selector.
    le.P_coag+P_flowin+P_flowout+P_evap) then
    coag=4
end if
426
428 ! write (*,*) "Total probability =", P_coag+P_flowin+P_flowout+
P_evap
! write (*,*) "Coag flag =", coag ! For debugging purposes
430 return
end subroutine event_selection

Appendix--A/event_selection.f95
A.6 Binder Flow Subroutine

```fortran
subroutine binder_flow

!***** Define global variables *****
use particle

!***** Define local variables *****
implicit none
common/r/iseed
integer holder(n_binder2)
real ran0
integer iseed
integer ii, jj, counter

!write(*,*) "Binder flow" ! For debugging purposes
binder_flag=0

if(n_binder1.ge.1.0)then
  binder_flag=1
  binder_selection=binder_selection+n_binder1
  go to 110
elseif(ran0(iseed).le.n_binder1) then ! Used to approximate the case
  ! where a "fraction" of a binder droplet enters with each event
  !write(*,*) "Binder particle selected."
  binder_flag=1
  binder_selection=binder_selection+1
  go to 110

!***** Pick n_binder2 number of particles to be replaced by binder particles *****
110 counter=1
  holder(1) = 1+n*ran0(iseed)
  if(holder(1).eq.n+1) then ! Needed to prevent a v(10,001) scenario
    holder(1)=n
  end if

if(n_binder2.ne.1)then
  do ii = 2, n_binder2
```
holder(ii)=holder(1) 

100 do jj = 1, counter 
if (holder(ii).eq. holder(jj)) then 
  holder(ii) = 1+n*ran0(iseed) 
  if(holder(ii).eq.n+1) then  ! Needed to prevent a v(10,001) scenario
    holder(ii)=n 
  end if
  ! write(*,*) "holder",(ii),"=", holder(ii)  ! For debugging purposes
  ! write(*,*) "holder",(jj),"=", holder(jj)  ! For debugging purposes
  go to 100
end if
end do 

counter = counter +1 
end do 
end if 

!**** Replace simulation particles with binder particles ****

55 do ii = 1, n_binder2 
  if (((m(holder(ii))+w(holder(ii)))/v(holder(ii)).eq.1.0) then 
    binder = binder -1 
    part = part +1 
  end if
  ! write (*,*) "Holder", ii, "," , holder(ii)  ! For debugging purposes
  v(holder(ii))=vb+wb 
  v_m(holder(ii))=vb+wb 
  rho_gran(holder(ii))=v_m(holder(ii))/v(holder(ii))
  v_1(holder(ii))=0.0d0 
  v_2(holder(ii))=0.0d0 
  powder1_mass_frac(holder(ii))=0.0d0 
  m(holder(ii))=vb 
  w(holder(ii))=wb 
  w_m(holder(ii))=wb 
  HPC_mass_frac(holder(ii))=m(holder(ii))/(m(holder(ii))+w_m(holder(ii)))
! write(*,*) "v", holder(ii), ",=", v(holder(ii)) ! For debugging purposes
! write(*,*) "m", holder(ii), ",=", m(holder(ii)) ! For debugging purposes
end do
end if

**** Call subroutine for updating calculations ****
call Calc_updates
return
end subroutine binder_flow
A.7 Coagulation Subroutine

```fortran
subroutine coagulation

!**** Define global variables ****
use particle

!**** Define local variables ****
implicit none
common/r/iseed
integer success, trial
integer i, j, k
real ran0
integer iseed
real*8 prob_coagulation, kernel, kmax, kernel2, kmax2

success=0 ! flag to indicate acceptance of coagulation pair
trial=0 ! number of trials till success
!write(*,*) ”Coagulation” ! For debugging purposes

!**** Pick a pair of particles ****
while(success.eq.0)
trial=trial+1
800 i = 1+n*ran0(iseed)
   if(i.eq.n+1) then ! Needed to prevent a v(10,001) scenario
      i=n
   end if
   j=i
   while (j.eq.i)
      j = 1+n*ran0(iseed)
      if(j.eq.n+1) then ! Needed to prevent a v(10,001) scenario
         j=i
      end if
      if((m(i)+w(i))/v(i).eq.1.0.and.(m(j)+w(j))/v(j).eq.1.0) then
         !write(*,*) ”j =”, j
         !write(*,*) ”c(i) =”, m(i)/v(i)
         !write(*,*) ”c(j) =”, m(j)/v(j)
         j=i
      end if
   go to 800
   end if
```
end do

!**** Calculate simple sigmoidal-correlation Psi factor ****
!! if (v(i).eq.(m(i)+w(i)).and.v(i).ne.m(i)) then
!! bs_ratio1 = 1.0
!! else
!! bs_ratio1 = w(i)/(v(i)-w(i)) ! Use if ratio is to be aq. binder content over dry granule content
!!! bs_ratio1 = (m(i)+w(i))/(v(i)-(m(i)+w(i))) ! Use if ratio is to be entire binder content over powder content
!! end if
!write(*,* ) "bs_ratio1 =", bs_ratio1
!! if (v(j).eq.(m(j)+w(j)).and.v(j).ne.m(j)) then
!! bs_ratio2 = 1.0
!! else
!! bs_ratio2 = w(j)/(v(j)-w(j)) ! Use if ratio is to be aq. binder content over dry granule content
!!! bs_ratio2 = (m(j)+w(j))/(v(j)-(m(j)+w(j))) ! Use if ratio is to be entire binder content over powder content
!! end if
!write(*,* ) "bs_ratio2 =", bs_ratio2
!! if (bs_ratio1.gt.0.735) then
!! frc1=1.0
!! end if
!! if (bs_ratio1.le.0.735) then
!! frc1=1.0291*((1.0-exp(-6.8539*bs_ratio1))**4.4089)
!! end if
!! if (bs_ratio1.eq.0.0) then
!! frc1=0.0
!! end if
!! if (bs_ratio2.gt.0.735) then
!! frc2=1.0
!! end if
!! if (bs_ratio2.le.0.735) then
!! $fr_c2 = 1.0291 * ((1.0 - \exp(-6.8539 * bs\_ratio2)) ** 4.4089)$
!! end if

!! if (bs\_ratio2 . eq. 0.0) then
!! $fr_c2 = 0.0$
!! end if

!* **** Calculation of geometrical $\Psi$ *****
!* **** Calculation of $\varepsilon$as *****

!* **** "a" values *****
if (powder\_flag1 . eq. 1) then
a1 = 1.0 ! Avicel value
elseif (powder\_flag1 . eq. 2) then
a1 = 1.0 ! A-TAB value
elseif (powder\_flag1 . eq. 3) then
a1 = 1.0 ! Lactose value
elseif (powder\_flag1 . eq. 4) then
a1 = 1.0 ! Mannitol value
end if

if (powder\_flag2 . eq. 1) then
a2 = 1.0 ! Avicel value
elseif (powder\_flag2 . eq. 2) then
a2 = 1.0 ! A-TAB value
elseif (powder\_flag2 . eq. 3) then
a2 = 1.0 ! Lactose value
elseif (powder\_flag2 . eq. 4) then
a2 = 1.0 ! Mannitol value
end if

a\_gran1 = powder1\_mass\_frac(i) * a1 + (1.0d0 - powder1\_mass\_frac(i)) * a2
a\_gran2 = powder1\_mass\_frac(j) * a1 + (1.0d0 - powder1\_mass\_frac(j)) * a2

!* **** "b" values *****
if (powder\_flag1 . eq. 1) then
b1 = 35.1 ! Avicel value
elseif (powder\_flag1 . eq. 2) then
b1 = 33.5 ! A-TAB value
elseif(powder_flag1.eq.3) then
  b1=33.0  ! Lactose value
elseif(powder_flag1.eq.4) then
  b1=27.4  ! Mannitol value
end if

if(powder_flag2.eq.1) then
  b2=35.1  ! Avicel value
elseif(powder_flag2.eq.2) then
  b2=33.5  ! A−TAB value
elseif(powder_flag2.eq.3) then
  b2=33.0  ! Lactose value
elseif(powder_flag2.eq.4) then
  b2=27.4  ! Mannitol value
end if

b_gran1=powder1_mass_frac(i)*b1+(1.0d0−powder1_mass_frac(i))*b2
b_gran2=powder1_mass_frac(j)*b1+(1.0d0−powder1_mass_frac(j))*b2

!***** "c" values *****
if(powder_flag1.eq.1) then
  c1=0.129  ! Avicel value
elseif(powder_flag1.eq.2) then
  c1=0.333  ! A−TAB value
elseif(powder_flag1.eq.3) then
  c1=0.206  ! Lactose value
elseif(powder_flag1.eq.4) then
  c1=0.182  ! Mannitol value
end if

if(powder_flag2.eq.1) then
  c2=0.129  ! Avicel value
elseif(powder_flag2.eq.2) then
  c2=0.333  ! A−TAB value
elseif(powder_flag2.eq.3) then
  c2=0.206  ! Lactose value
elseif(powder_flag2.eq.4) then
  c2=0.182  ! Mannitol value
end if
c_gran1=powder1_mass_frac(i)*c1+(1.0d0-powder1_mass_frac(i))*c2
c_gran2=powder1_mass_frac(j)*c1+(1.0d0-powder1_mass_frac(j))*c2

if (v(i).eq.(m(i)+w(i)).and.v(i).ne.m(i)) then
  bs_ratio1 = 1.0
else
  !!! bs_ratio1 = w(i)/(v(i)-w(i)) ! Use if ratio is to be aq. binder content over dry granule content
  bs_ratio1 = (m(i)+w(i))/(v(i)-(m(i)+w(i))) ! Use if ratio is to be entire binder content over powder content
end if

if (v(j).eq.(m(j)+w(j)).and.v(j).ne.m(j)) then
  bs_ratio2 = 1.0
else
  !!! bs_ratio2 = w(j)/(v(j)-w(j)) ! Use if ratio is to be aq. binder content over dry granule content
  bs_ratio2 = (m(j)+w(j))/(v(j)-(m(j)+w(j))) ! Use if ratio is to be entire binder content over powder content
end if

eta1=a_gran1/(1.0+EXP(-b_gran1*(bs_ratio1-c_gran1)))
eta2=a_gran2/(1.0+EXP(-b_gran2*(bs_ratio2-c_gran2)))

! write(*,* )"eta1 = ", eta1 ! used for debugging purposes
! write(*,* )"eta2 = ", eta2 ! used for debugging purposes

if(eta1.gt.1.0) then
  write(*,* )" !!!!! Eta1 > 1.0 !!!!!"
end if

if(eta2.gt.1.0) then
  write(*,* )" !!!!! Eta2 > 1.0 !!!!!"
end if

geom_Psi = 1 - (1-eta1)*(1-eta2)

!**** Calculation of physical Psi ****
!**** Calculation of Stokes number ****

St=(4.0/9.0)*((v(i)**(1.0/3.0))*(v(j)**(1.0/3.0))+(v(i)**(1.0/3.0))*((v(i)**(1.0/3.0))+v(j))**2.0)/(v(i)+v(j))
\[
St = \frac{(4.0/9.0) \times (\text{u/mu}) \times (\rho_{\text{gran}}(i) \times \rho \times \rho_{\text{gran}}(j) \times \rho) \times d_0 \times ((v(i)^{1.0/3.0}) \times (v(j)^{1.0/3.0})) \times 2.0}{(\rho_{\text{gran}}(i) \times \rho \times v(i) + \rho_{\text{gran}}(j) \times \rho \times v(j))}
\]

! write(*,*) "v(i) = ", v(i)
! write(*,*) "v(j) =", v(j)
! write(*,*) "St =", St

***** Calculation of critical Stokes number *****

***** Calculation of lambda *****

***** Calculation of relative displaced binder volume, phi13 ****

***** "f" values *****
if (powder_flag1.eq.1) then
  f1 = 25.9 ! Avicel value
elseif (powder_flag1.eq.2) then
  f1 = 30.4 ! A−TAB value
elseif (powder_flag1.eq.3) then
  f1 = 29.8 ! Lactose value
elseif (powder_flag1.eq.4) then
  f1 = 26.8 ! Mannitol value
end if

if (powder_flag2.eq.1) then
  f2 = 25.9 ! Avicel value
elseif (powder_flag2.eq.2) then
  f2 = 30.4 ! A−TAB value
elseif (powder_flag2.eq.3) then
  f2 = 29.8 ! Lactose value
elseif (powder_flag2.eq.4) then
  f2 = 26.8 ! Mannitol value
end if

f_gran1 = powder1_mass_frac(i) \times f1 + (1.0d0 - powder1_mass_frac(i)) \times f2
f_gran2 = powder1_mass_frac(j) \times f1 + (1.0d0 - powder1_mass_frac(j)) \times f2

***** "g" values *****
if (powder_flag1.eq.1) then
if (powder_flag1 . eq . 1) then
  g1 = 0.327  ! Avicel value
else if (powder_flag1 . eq . 2) then
  g1 = 0.538  ! A-TAB value
else if (powder_flag1 . eq . 3) then
  g1 = 0.403  ! Lactose value
else if (powder_flag1 . eq . 4) then
  g1 = 0.394  ! Mannitol value
end if

if (powder_flag2 . eq . 1) then
  g2 = 0.327  ! Avicel value
else if (powder_flag2 . eq . 2) then
  g2 = 0.538  ! A-TAB value
else if (powder_flag2 . eq . 3) then
  g2 = 0.403  ! Lactose value
else if (powder_flag2 . eq . 4) then
  g2 = 0.394  ! Mannitol value
end if

g_gran1 = powder1_mass_frac(i) * g1 + (1.0d0 - powder1_mass_frac(i)) * g2
g_gran2 = powder1_mass_frac(j) * g1 + (1.0d0 - powder1_mass_frac(j)) * g2

phi13_1 = EXP( f_gran1 * (bs_ratio1 - g_gran1) )
phi13_2 = EXP( f_gran2 * (bs_ratio2 - g_gran2) )

lambda1 = rg * (phi13_1 ** (1.0/3.0))
lambda2 = rg * (phi13_2 ** (1.0/3.0))
lambda = lambda1 + lambda2
! write(*,*), "lambda = ", lambda ! used for debugging purposes

!***** Values for ha (input data in micrometers) *****
if (powder_flag1 . eq . 1) then
  ha1 = 6.3  ! Avicel value
else if (powder_flag1 . eq . 2) then
  ha1 = 16.5  ! A-TAB value
else if (powder_flag1 . eq . 3) then
  ha1 = 10.5  ! Lactose value
elseif ( powder_flag1.eq.4 ) then
        ha1=10.5 ! Mannitol value
end if

if ( powder_flag2.eq.1 ) then
        ha2=6.3 ! Avicel value
elseif ( powder_flag2.eq.2 ) then
        ha2=16.5 ! A-TAB value
elseif ( powder_flag2.eq.3 ) then
        ha2=10.5 ! Lactose value
elseif ( powder_flag2.eq.4 ) then
        ha2=10.5 ! Mannitol value
end if

ha_gran1=powder1_mass_frac(i)*ha1+(1.0d0-powder1_mass_frac(i))*ha2
ha_gran2=powder1_mass_frac(j)*ha1+(1.0d0-powder1_mass_frac(j))*ha2

ha=(ha_gran1+ha_gran2)/2.0d0

critical_St=2.0*log(lambda/ha)

if (St.le.critical_St) then
        phys_Psi=1.0
else
        phys_Psi=0.0
end if

! write(*,*) "Physical Psi =", phys_Psi

! Warning statement
! if (phys_Psi.eq.0) then
!     write(*,*) "!!!!!!!!!!!!! Physical Psi = 0
!!!!!!!!!!!!!!!!!"
! end if

***** Calculation of freshness Psi *****
if (HPC_mass_frac(i).ge.0.75.and.HPC_mass_frac(j).ge.0.75) then
        fresh_Psi=0.0
else
fresh_Psi=1.0
end if

! fresh_Psi=1.0 ! Used when you do not wish to consider a
freshness Psi factor

**** Calculation of Psi (Collision success factor) ****
Psi=1.0
! For geometrical Psi:
!!!Psi=1.0−(1.0−frc1)*(1.0−frc2)
! For Stokes based Psi:
Psi=geom_Psi*phys_Psi*fresh_Psi
! write (*,*) "Psi =", Psi

if((m(i)+w(i))/v(i).ne.1.0.and.(m(j)+w(j))/v(j).ne.1.0) then
prob_coagulation=kernel(i,j)/kmax(0)
! write (*,*) "kmax =", kmax(0) ! Used for debugging purposes
! write (*,*) "kernel =", kernel(i,j) ! Used for debugging purposes
else
prob_coagulation=kernel2(i,j)/kmax2(0)
! write (*,*) "kmax2 =", kmax2(0) ! Used for debugging purposes
! write (*,*) "kernel2 =", kernel2(i,j) ! Used for debugging purposes
end if

! write (*,*) "trial =", trial ! Used for debugging purposes
! write (*,*) "Prob_coagulation =", prob_coagulation ! Used for debugging purposes
! write (*,*) "kmax =", kmax(0) ! Used for debugging purposes
! write (*,*) "kernel =", kernel(i,j) ! Used for debugging purposes
! write (*,*) "v(i) =", v(i) ! Used for debugging purposes
! write (*,*) "v(j) =", v(j) ! Used for debugging purposes
! write (*,*) "imax =", maxval(v) ! Used for debugging purposes
! write (*,*) "imin =", minval(v) ! Used for debugging purposes
! write (*,*) "prob =", prob_coagulation
if(ran0(iseed).lt.prob_coagulation) then
success=1

**** Calculate change in number of binder and powder particles
if((m(i)+w(i))/v(i).eq.1.0.and.(m(j)+w(j))/v(j).ne.1.0) then
  binder = binder - 1
end if
if((m(i)+w(i))/v(i).ne.1.0.and.(m(j)+w(j))/v(j).eq.1.0) then
  binder = binder - 1
end if
if((m(i)+w(i))/v(i).ne.1.0.and.(m(j)+w(j))/v(j).ne.1.0) then
  part = part - 1
end if

**** Write warning if calculated kernel is greater than the kmax value
if(kernel(i,j).gt.kmax(0).or.kernel2(i,j).gt.kmax2(0)) then
  write(*,*),("Iinsufficient Kmax")
  rr=rr+1
end if

end if
end do

**** Pick k to replace j ****
k=j
while (k.eq.j)
  k = 1+n*ran0(iseed)
  if(k.eq.n+1) then ! Needed to prevent a v(10,001) scenario
    k=n
  end if
end do

! call wrt_before(i, j, k) ! (debug) uncomment if you want to print
Perform coagulation event

\[ v(i) = v(i) + v(j) \]
\[ v_m(i) = v_m(i) + v_m(j) \]
\[ \rho_{\text{gran}}(i) = v_m(i) / v(i) \]
\[ v_1(i) = v_1(i) + v_1(j) \]
\[ v_2(i) = v_2(i) + v_2(j) \]
\[ m(i) = m(i) + m(j) \]
\[ w(i) = w(i) + w(j) \]
\[ w_m(i) = w_m(i) + w_m(j) \]
\[ \text{powder1_mass_frac}(i) = v_1(i) / (v_1(i) + v_2(i)) \]
\[ \text{HPC_mass_frac}(i) = m(i) / (m(i) + w_m(i)) \]
\[ v(j) = v(k) \]
\[ v_m(j) = v_m(k) \]
\[ \rho_{\text{gran}}(j) = v_m(j) / v(j) \]
\[ v_1(j) = v_1(k) \]
\[ v_2(j) = v_2(k) \]
\[ m(j) = m(k) \]
\[ w(j) = w(k) \]
\[ w_m(j) = w_m(k) \]
\[ \text{if} \ (1.0 * (m(j) + w(j)) / v(j) < 1.0) \text{ then} \]
\[ \text{powder1_mass_frac}(j) = v_1(j) / (v_1(j) + v_2(j)) \]
\[ \text{else} \]
\[ \text{powder1_mass_frac}(j) = 0.0 \]
\[ \text{end if} \]
\[ \text{if} \ (m(j) \neq 0.0) \text{ then} \]
\[ \text{HPC_mass_frac}(j) = 0.0 \]
\[ \text{else} \]
\[ \text{HPC_mass_frac}(j) = m(j) / (m(j) + w_m(j)) \]
\[ \text{end if} \]
\[ \text{if} \ ((m(j) + w(j)) / v(j) \neq 1.0) \text{ then} \]
\[ \text{binder} = \text{binder} + 1 \]
\[ \text{else} \]
\[ \text{part} = \text{part} + 1 \]
\[ \text{end if} \]

Update parameters for calculation of \( k_{\text{max}} \)
\[ \text{if} \ (v(i) > v_{\text{max}}) \text{ then} \]
409  vmax=v(i)  
   end if
411  ! write(*,*) 'v(i) =', v(i) ! Used for debugging purposes
   ! write(*,*) 'max particle mass =', vmax ! Used for debugging purposes
413  ! write(*,*) 'min particle mass =', vmin ! Used for debugging purposes
415  if ((m(i)+w(i)).gt.mmax) then  
      mmax=m(i)+w(i)  
   end if
417  ! write(*,*) 'm(i) =', m(i) ! Used for debugging purposes
   ! write(*,*) 'max solute mass =', mmax ! Used for debugging purposes
   ! write(*,*) 'min solute mass =', mmin ! Used for debugging purposes
421  ! Call subroutine for updating calculations
423  call Calc_updates
425  ! call wrt_after(i, j, k) ! (debug) uncomment if you want to print
return
427  end subroutine coagulation
A.8  Evaporation Subroutine

subroutine evaporation

!*****  Define global variables *****
use particle

!*****  Define local variables *****
implicit none
common/r/iseed
integer success, trial
integer i, j, k
integer iseed
real ran0

success=0 ! flag to indicate acceptance of particle for evaporation
trial=0 ! number of trials till success

!*****  Pick particle for evaporation event *****
while(success.eq.0)
  trial=trial+1
  i = 1+n*ran0(iseed)
  if(i.eq.n+1) then ! Needed to prevent a v(10,001) scenario
    i=n
  end if
  !if(loop.gt.310) then ! Used for debugging purposes
  !write(*,*) "trial =", trial ! Used for debugging purposes
  !end if ! Used for debugging purposes
  if(w_m(i).ge.evap_mass) then
    success=1
  end if
end do

call wrt_before(i, j, k) ! (debug) uncomment if you want to print

!*****  Perform evaporation event *****
v(i)=v(i)
v_m(i)=v_m(i)-evap_mass
rho_gran(i)=v_m(i)/v(i)
v_1(i)=v_1(i)
v_2(i)=v_2(i)
m(i)=m(i)
w(i)=w(i)
w_m(i)=w_m(i)-evap_mass
HPC_mass_frac(i)=m(i)/(m(i)+w_m(i))

!***** Call subroutine for updating calculations *****
call Calc_updates
!call wrt_after(i,j,k) ! (debug) uncomment if you want to print
return
end subroutine evaporation

Appendix–A/evaporation.f95
A.9  Calculation Update Subroutine

```fortran
subroutine Calc_updates

!***** Define global variables *****
use particle

!***** Define local variables *****
implicit none
common/r/iseed
integer success, trial
integer i, j, k
real ran0
integer iseed
integer ii, jj
real*8 prob_coagulation, kernel, kmax
real*8 kernelsum
real*8 Vavg ! parameter used to calculate average mass
real*8 dv_event ! dimensionless mass change due to the selected event
real*8 dv_moist
real*8 delta_mass_conc
real*8 delta_moist
real*8 t_denom, t_denom1, t_denom2
real*8 t_term2

!***** Determine new dimensionless mass concentration of particles *****
    dv_event=0.0
    if(coag.eq.2.or.coag.eq.4) then
        if(coag.eq.2.and.binder_flag.eq.1) then
            dv_event=(1.0*n_binder2)*(vb+wb) ! Particle mass change assoc
                ! with binder flow in
        end if
    end if

    if(coag.eq.4) then
        dv_event=-evap_mass
        !dv_event=0.0
    end if
```

delta_mass_conc = mass_conc*dv_event/(1.0*n)/vav
mass_conc = mass_conc+delta_mass_conc
/write(*,*) "dv event =", dv_event ! Used for debugging purposes
/write(*,*) "Delta mass concentration =", delta_mass_conc ! Used for debugging purposes
/write(*,*) "New mass concentration =", mass_conc ! Used for debugging purposes
end if

!* * * * * Determine new dimensionless mass concentration of moisture * * * * *
*dv_moist=0.0
if(coag.eq.2.or.coag.eq.3.or.coag.eq.4) then
  if(coag.eq.2) then
dv_moist=n_binder1*cw0/(Cn0/C0) ! Moisture assoc. with flow in
  end if

  if(coag.eq.3) then
dv_moist=-epsilon_out*moisture
  end if

end if

if(coag.eq.4) then
dv_moist=evap_mass
end if

delta_moist = mass_conc*dv_moist/(1.0*n)/vav
moisture = moisture+delta_moist
if(moisture.lt.0.0) then
  moisture=0.0
end if

!***** Write if statement to ensure that the moisture isn’t greater than the saturation moisture. *****
if(moisture.ge.(1.0d0-vb)*cw_sat) then
  moisture=(1.0d0-vb)*cw_sat
end if
/write(*,*) "dv moist =", dv_moist ! Used for debugging purposes
! write(*,*) "Delta moisture concentration =", delta_moist ! Used for debugging purposes
! write(*,*) "New moisture mass concentration =", moisture ! Used for debugging purposes
end if

!**** Determine after event average mass ****
mass=0
solute=0
do ii=1, n
  !mass=mass+v(ii) ! Use when moisture is not removed from system
  mass=mass+v*m(ii)
  !solute=solute+m(ii) ! Use only if solute is to include dry binder content only
  !solute=solute+m(ii)+w(ii) ! Use if solute is to include both aq. and dry portions of binder
  solute=solute+m(ii)+w*m(ii)
end do

vav=mass/(1.0*n)
! write(*,*) "After Vav =", vav ! Used for debugging purposes

!**** Determine new dimensionless number concentration ****
conc=mass_conc/vav
! write(*,*) "New number concentration =", conc ! Used for debugging purposes

!**** For calculation of evaporation rate ****
eta_counter=1

eta_holder(1) = 1+n*ran0(iseed)
if(eta_holder(1).eq.n+1.or.v(eta_holder(1)).eq.1.0d0) then ! Needed to prevent a v(10,001) scenario
  eta_holder(1)=n
end if
do ii = 2, eta_number
  eta_holder(ii)=eta_holder(1)
do jj = 1, eta_counter
  if (eta_holder(ii).eq.eta_holder(jj)) then
    eta_holder(ii) = 1+n*ran0(iseed)
  if(eta_holder(ii).eq.n+1.or.v(eta_holder(ii)).eq.1.0d0) then
    ! Needed to prevent a v(10,001) scenario
    eta_holder(ii)=n
  end if
  !write(*,*) "eta_holder",(ii)," =", eta_holder(ii) ! For debugging purposes
  !write(*,*) "eta_holder",(jj)," =", eta_holder(jj) ! For debugging purposes
  go to 135
end if
end do

! if(ii.eq.eta_number) then ! For debugging purposes
! do jj=1,eta_number ! For debugging purposes
! write(*,*) "eta_holder ", jj, " = ", eta_holder(jj) ! For debugging purposes
! end do ! For debugging purposes
! end if ! For debugging purposes
eta_counter = eta_counter +1
end do

ap_sum=0.0
drymass=0.0
moisturemass=0.0
do ii=1,eta_number
  ap_sum=ap_sum+(v(eta_holder(ii))**2.0/3.0)
  drymass=drymass+v(eta_holder(ii))-w(eta_holder(ii))
  !!moisturemass=moisturemass+w(eta_holder(ii))
  moisturemass=moisturemass+w_m(eta_holder(ii))
end do

average_ap=ap_sum/(1.0d0*eta_number)
LOD_dry=moisturemass/drymass

**** Evaluate time interval for each step ****
t_denom1=(average_kernel*(conc**2.0)/2.0)+(tauc_tauin/epsilon_in)
t_denom2=(tauc_tauout/epsilon_out)+(tauc_taeuvap*LOD_dry*average_ap
*(1.0d0-vb)*cw_sat-moisture)*conc/evap_mass) ! Rajniak new
empirical evaporation rate
!! t_term2=(tauc_taeuvap*LOD_dry*average_ap*((1.0d0-vb)*cw_sat-
moisture)*conc/evap_mass)
!! t_denom2=0.0
139 if(moist_count.eq.0) then
   t_denom2=(tauc_tauout/epsilon_out)
end if
141 t_denom=t_denom1+t_denom2
145 !t_denom=(tauc_taeuvap*(cw_sat-moisture)/evap_mass) ! For
   evaporation only
deltat=conc/(1.0*n)/t_denom
147 time=time+deltat
149 ! write (*,*) "deltat =", deltat ! Used for debugging purposes
! write (*,*) "time =", time ! Used for debugging purposes
151 return
end subroutine Calc_updates

Appendix–A/Calc_updates.f95
A.10 Particle-Particle Kernel Calculation

```fortran
!--- file provides two functions:
! kernel: kernel as a function of particles i, j
! kmax : max value of kernel

function kernel(i,j)
use particle
implicit none
real*8 kernel
real*8 k0
real*8 term1
real*8 term2
real*8 term3
real*8 comp
integer i, j
character(80) kernel_type ! string that identifies the kernel

common/k/kernel_type
kernel_type="KTGF kernel/composition dependent"
k0=K_PP/K_PP
! aAA=-3.0
! aAB=-2.0
! aBB=+2.0
comp=0.0
! comp=-aAB*(m(i)/v(i))*(1.0-(m(j)/v(j)))+(m(j)/v(j))*(1.0-(m(i)/v(i)))
! term1=(aAA*(m(i)/v(i))*(m(j)/v(j))
! term2=(aBB*(1.0-(m(i)/v(i)))*(1.0-(m(j)/v(j))))
! term3=(aAB*((m(i)/v(i))*(1.0-(m(j)/v(j)))+(m(j)/v(j))*(1.0-(m(i)/v(i))))
! comp=-term1-term2-term3
if(d43_gamma_flag.eq.2) then
  kernel=k0*Psi(((v(i)**(1.0/3.0))+(v(j)**(1.0/3.0)))*2.0)
  *(((1.0/v(i))+v(j))+(1.0/v(j)))*exp(comp)
end if
if(d43_gamma_flag.eq.1) then
```

207
kernel=k0*Psi*(x43**gamma)*(((v(i)**(1.0/3.0))+(v(j)**(1.0/3.0)))*
**2.0)*(((1.0/v(i))+(1.0/v(j)))*(1.0/2.0))*exp(comp)
!!kernel=k0*Psi*(x43**gamma)*(((v(i)**(1.0/3.0))+(v(j)**(1.0/3.0)))*
**2.0)*(((1.0/(rho_gran(i)*v(i)))+(1.0/(rho_gran(j)*v(j))))
***(1.0/2.0))*exp(comp)
end if
! kernel=k0*exp(comp) ! For constant kernel
return
end

function kmax(ii)
!--------------------------------------------------------------------------
! This function calculates the MAXIMUM kernel in the sim box.
! For the composition independent KTGF kernel, this is between
! two particles—one with maximum mass and the other with
! minimum mass (other kernels may require a different
! calculation). The exact value of kmax is *not* critical as
! long as the estimate of kmax is not smaller than the maximum
! kernel in the simulation box.
!--------------------------------------------------------------------------
! IMPORTANT
! this function must be called using
! a dummy argument that is declared an
! integer. Example: kijmax(0)
!--------------------------------------------------------------------------
use particle
implicit none
!real*8 kernel
real*8 kmax,k0max
!real*8 terma
!real*8 termb
!real*8 termc
real*8 comp
integer ii
if(KPB.gt.K_PP) then
  k0max=K_PB/K_PP
else
  k0max=1.0
end if
! aAA=−3.0
! aAB=−2.0
! aBB=+2.0

! For KTGF kernel and negative aAB values, use the kmax immediately below.
comp=0.0
! comp=aAB*((mmax/mmax)*(1.0−(mmin/vmin))+(mmin/vmin)*(1.0−(mmax/mmax)))

! terma=aAA*1*1 ! Used in CERD paper Cases 4–6c
! termb=aBB*0*0 ! Used in CERD paper Cases 4–6c
! termc=aAB*(1*0+1*0) ! Used in CERD paper Cases 4–6c
! comp=−terma−termb−termc

if (d43.gamma_flag.eq.2) then
  kmax=k0max*(((vmax**(1.0/3.0))+(vmin**(1.0/3.0)))**2.0)*(((1.0/vmax)+(1.0/vmin))**2.0)*exp(comp)
end if

if (d43.gamma_flag.eq.1) then
  kmax=k0max*(x43**gamma)*(((vmax**(1.0/3.0))+(vmin**(1.0/3.0)))**2.0)*(((1.0/vmax)+(1.0/vmin))**2.0)*exp(comp)
end if

! For KTGF kernel and positive aAB values, use the kmax immediately below.
! kmax=(k0max*(((vmax**(1.0/3.0))+(vmin**(1.0/3.0)))**2.0)*(((1.0/vmax)+(1.0/vmin))**2.0))+1

! For constant kernel and negative aAB values, use the kmax immediately below.
! comp=0.0
! comp=aAB*((mmax/mmax)*(1.0−(mmin/vmin))+(mmin/vmin)*(1.0−(mmax/mmax)))

! terma=aAA*1*1
! termb=aBB*0*0
! termc=aAB*(1*0+1*0)
! comp=−terma−termb−termc

! kmax=k0max*exp(comp)
For constant kernel and positive aAB values, use the kmax immediately below.

! kmax=kernel

! kmax=1.d0*kernel(vmax,vmax)

return

end

Appendix-A/KTGF_comp_depend_kernel.f95
A.11 Particle-Binder Droplet Kernel Calculation

```fortran

! file provides two functions:
! kernel: kernel as a function of particles i, j
! kmax : max value of kernel

function kernel2(i,j)
use particle
implicit none
real*8 kernel2
real*8 k02
real*8 term12
real*8 term22
real*8 term32
real*8 comp2
integer i, j
character*(80) kernel_type2 ! string that identifies the kernel
common/k/kernel_type2
kernel_type2="KTGF kernel/composition dependent"
k02=K_PB/K_PP
aAA2=-3.0
aAB2=-2.0
aBB2=+2.0
comp2=0.0

! comp2=-aAB2*((m(i)/v(i))*(1.0-(m(j)/v(j)))+(m(j)/v(j))*(1.0-(m(i)/v(i))))
! term12=(aAA2*(m(i)/v(i))*(m(j)/v(j)))
! term22=(aAB2*(1.0-(m(i)/v(i)))*(1.0-(m(j)/v(j))))
! term32=(aBB2*((m(i)/v(i))*(1.0-(m(j)/v(j)))+(m(j)/v(j))*(1.0-(m(i)/v(i)))))

! comp2=term12-term22-term32
if(d43_gamma_flag.eq.2) then
  kernel2=k02*((v(i)**(1.0/3.0)+(v(j)**(1.0/3.0))**2.0)** (((1.0/v(i))+(1.0/v(j)))*(1.0/2.0))**exp(comp2)
end if
if(d43_gamma_flag.eq.1) then
```

211
kernel2=k02*(x43*gamma)*(((v(i)**(1.0/3.0))+(v(j)**(1.0/3.0)))
**2.0)*(((1.0/v(i))+(1.0/v(j)))**(1.0/2.0))*exp(comp2)

end if

! kernel2=k02*exp(comp2)  ! For constant kernel
return
end

function kmax2(ii)

! This function calculates the MAXIMUM kernel in the sim box.
! For the composition independent KTGF kernel, this is between
! two particles—one with maximum mass and the other with
! minimum mass (other kernels may require a different
! calculation). The exact value of kmax is *not* critical as
! long as the estimate of kmax is not smaller than the maximum
! kernel in the simulation box.

! IMPORTANT
! this function must be called using
! a dummy argument that is declared an
! integer. Example: kijmax(0)

use particle
implicit none
! real*8 kernel2
real*8 kmax2,k0max
! real*8 terma2
! real*8 termb2
! real*8 termc2
real*8 comp2
integer ii
if(K_PB.gt.K_PP) then
   k0max=K_PB/K_PP
else
   k0max=1.0
end if
$a_{AA2} = -3.0$

$\tilde{a}_{BB2} = +2.0$

For KTGF kernel and negative $a_{AB}$ values, use the $k_{max}$ immediately below.

\[
\text{comp}_2 = 0.0
\]

For KTGF kernel and negative $a_{AB}$ values, use the $k_{max}$ immediately below.

\[
\text{comp}_2 = a_{AB2} \cdot \left( (\frac{m_{max}}{m_{max}}) \cdot (1.0 - (\frac{m_{min}}{v_{min}})) + (\frac{m_{min}}{v_{min}}) \cdot (1.0 - (\frac{m_{max}}{m_{max}})) \right)
\]

\[
terma_2 = a_{AA2} \cdot 1 \ast 1 \quad \text{Used in CERD paper Cases 4-6c}
\]

\[
term_b_2 = a_{BB2} \cdot 0 \ast 0 \quad \text{Used in CERD paper Cases 4-6c}
\]

\[
termc_2 = a_{AB2} \cdot (1 \ast 0 + 1 \ast 0) \quad \text{Used in CERD paper Cases 4-6c}
\]

\[
\text{comp}_2 = \text{terma}_2 - \text{term}_b_2 - \text{termc}_2
\]

if \(d_{43}.gamma\_flag\) . eq . 2 then

\[
k_{max2} = k_{0max} \ast ((v_{max} \ast (1.0 / 3.0)) + (v_{min} \ast (1.0 / 3.0))) \ast (1.0 / v_{max}) + (1.0 / v_{min})) \ast \exp(\text{comp}_2)
\]

end if

if \(d_{43}.gamma\_flag\) . eq . 1 then

\[
k_{max2} = k_{0max} \ast (x_{43} \ast gamma) \ast ((v_{max} \ast (1.0 / 3.0)) + (v_{min} \ast (1.0 / 3.0))) \ast (1.0 / v_{max}) + (1.0 / v_{min})) \ast \exp(\text{comp}_2)
\]

end if

For KTGF kernel and positive $a_{AB}$ values, use the $k_{max}$ immediately below.

\[
k_{max2} = (k_{0max} \ast ((v_{max} \ast (1.0 / 3.0)) + (v_{min} \ast (1.0 / 3.0))) \ast (1.0 / v_{max}) + (1.0 / v_{min})) \ast (1.0 / 2.0)) + 1
\]

For constant kernel and negative $a_{AB}$ values, use the $k_{max}$ immediately below.

\[
\text{comp}_2 = 0.0
\]

\[
\text{comp}_2 = a_{AB2} \cdot \left( (\frac{m_{max}}{m_{max}}) \cdot (1.0 - (\frac{m_{min}}{v_{min}})) + (\frac{m_{min}}{v_{min}}) \cdot (1.0 - (\frac{m_{max}}{m_{max}})) \right)
\]

\[
terma_2 = a_{AA2} \cdot 1 \ast 1
\]

\[
term_b_2 = a_{BB2} \cdot 0 \ast 0
\]

\[
termc_2 = a_{AB2} \cdot (1 \ast 0 + 1 \ast 0)
\]

\[
\text{comp}_2 = \text{terma}_2 - \text{term}_b_2 - \text{termc}_2
\]

\[
k_{max2} = k_{0max} \ast \exp(\text{comp}_2)
\]

For constant kernel and positive $a_{AB}$ values, use the $k_{max}$ immediately below.
<table>
<thead>
<tr>
<th>Line</th>
<th>Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>104</td>
<td><code>! kmax2=kernel2</code></td>
</tr>
<tr>
<td></td>
<td><code>! kmax2=1.d00*kernel2(vmax,vmax)</code></td>
</tr>
<tr>
<td>106</td>
<td><code>return</code></td>
</tr>
<tr>
<td></td>
<td><code>end</code></td>
</tr>
</tbody>
</table>

Appendix-A/ KTGF_comp_depend_kernel2.f95
A.12 Calculation of Distribution Statistics Sub-routine

subroutine stats

**** Define global variables ****
use particle

**** Define local variables ****
implicit none
real*8 x ! variable for calc of indiv excess solute
real*8 xsum ! variable for calc of sum of excess solute
real*8 x2 ! variable for calc of individual variance
real*8 x2sum ! variable for calc of sum of indiv. variances
real*8 wetmass ! variable for calc of wet granule only vavg
real*8 wetvavg ! variable for calc of wet granule only vavg
real*8 dryvavg ! variable for calc of dry granule only vavg
real*8 dried_gran_mass ! variable for calc of dried granule vavg
real*8 dried_gran_vavg ! variable for calc of dried granule vavg
real*8 dried_v43
real*8 dried_d43
real*8 coag_rate
real*8 binder_rate
real*8 rate_ratio
real*8 LOD_wet
integer ii, jj

**** Set initial parameter values ****
mass=0.0
solute=0
x2sum=0.0
xsum=0.0
wetmass=0.0
drymass=0.0
dried_gran_mass=0.0
dried_v43=0.0
moisturemass=0.0
**Calculate various statistical quantities**

```fortran
! mass = mass + v(i)
mass = mass + v_m(i)
!
!! solute = solute + m(i) + w(i)
!! solute = solute + m(i) + w_m(i)
!
if (1.0d0 * (m(i) + w(i))/v(i).lt.1.0) then
  wetmass = wetmass + v(i)
endif
!
! Calculate eta_ap using simple sigmoidal-correlation Psi factor
!! if (v(i).eq.(m(i) + w(i)).and.v(i).ne.m(i)) then
!! bs_ratio1 = 1.0
!! else
!! bs_ratio1 = w(i)/(v(i) - w(i)) ! Use if ratio is to be aq. binder content over dry granule content
!! bs_ratio1 = (m(i) + w(i))/((v(i) - (m(i) + w(i)))) ! Use if ratio is to be entire binder content over powder content
!! end if
!
!! if (bs_ratio1.gt.0.735) then
!! frc1 = 1.0
```
!! end if

!! if (bs_ratio1.le.0.735) then
  !! frct1 = 1.0291*(1.0-exp(-6.8539*bs_ratio1))**4.4089)
!! end if

!! if (bs_ratio1.eq.0.0) then
  !! frct1 = 0.0
!! end if

!! eta_sum = eta_sum + frct1
!! eta_ap_sum = eta_ap_sum + (frct1*(v(ii)**(2.0/3.0)))

***** Calculate eta_ap using Stokes-based Psi factor *****
***** "a" values *****
if (powder_flag1.eq.1) then
  a1 = 1.0  ! Avicel value
elseif (powder_flag1.eq.2) then
  a1 = 1.0  ! A-TAB value
elseif (powder_flag1.eq.3) then
  a1 = 1.0  ! Lactose value
elseif (powder_flag1.eq.4) then
  a1 = 1.0  ! Mannitol value
end if

if (powder_flag2.eq.1) then
  a2 = 1.0  ! Avicel value
elseif (powder_flag2.eq.2) then
  a2 = 1.0  ! A-TAB value
elseif (powder_flag2.eq.3) then
  a2 = 1.0  ! Lactose value
elseif (powder_flag2.eq.4) then
  a2 = 1.0  ! Mannitol value
end if

a_gran1 = powder1_mass_frac(ii)*a1 + (1.0-d0-powder1_mass_frac(ii))*a2

***** "b" values *****
if (powder_flag1.eq.1) then
b1=35.1  ! Avicel value
elseif(powder_flag1.eq.2) then
  b1=33.5  ! A–TAB value
elseif(powder_flag1.eq.3) then
  b1=33.0  ! Lactose value
elseif(powder_flag1.eq.4) then
  b1=27.4  ! Mannitol value
end if

if(powder_flag2.eq.1) then
  b2=35.1  ! Avicel value
elseif(powder_flag2.eq.2) then
  b2=33.5  ! A–TAB value
elseif(powder_flag2.eq.3) then
  b2=33.0  ! Lactose value
elseif(powder_flag2.eq.4) then
  b2=27.4  ! Mannitol value
end if

b_gran1=powder1_mass_frac(ii)*b1+(1.0d0–powder1_mass_frac(ii))*b2

!***** "c" values *****
if(powder_flag1.eq.1) then
  c1=0.129  ! Avicel value
elseif(powder_flag1.eq.2) then
  c1=0.333  ! A–TAB value
elseif(powder_flag1.eq.3) then
  c1=0.206  ! Lactose value
elseif(powder_flag1.eq.4) then
  c1=0.182  ! Mannitol value
end if

if(powder_flag2.eq.1) then
  c2=0.129  ! Avicel value
elseif(powder_flag2.eq.2) then
  c2=0.333  ! A–TAB value
elseif(powder_flag2.eq.3) then
  c2=0.206  ! Lactose value
elseif(powder_flag2.eq.4) then
  c2=0.182  ! Mannitol value
end if

c_gran1=powder1_mass_frac(ii)*c1+(1.0d0−powder1_mass_frac(ii))*c2

if (v(ii).eq.(m(ii)+w(ii)).and.v(ii).ne.m(ii)) then
  bs_ratio1 = 1.0
else
  bs_ratio1 = w(ii)/(v(ii)−w(ii))  ! Use if ratio is to be
  ! aq. binder content over dry granule content
  ! ! ! ! bs_ratio1 = (m(ii)+w(ii))/(v(ii)−(m(ii)+w(ii))) ! Use if
  ! ratio is to be entire binder content over powder content
end if

eta1=a_gran1/(1.0+EXP(−b_gran1*(bs_ratio1−c_gran1)))
eta_sum=eta_sum+eta1
eta_ap_sum=eta_ap_sum+(eta1*(v(ii)**(2.0/3.0)))
end do

!### Calculate various statistical quantities ###

vav=mass/(1.0*n)
phi=solute/(1.0d00*mass)
wetvavg=wetmass/(1.0*jj)
dryvavg=drymass/(1.0*jj)
dried_d43 = (dried_v43/drymass)*d0*1.0d6
!!! run_dried_d43(hh)=dried_d43

!!! run_time(hh)=time

LOD_wet=moisturemass/wetmass
LOD_dry=moisturemass/drymass
average_eta=eta_sum/(1.0*n)
average_eta_ap=eta_ap_sum/(1.0*n)

do ii=1,n
  x=(m(ii)−(phi*v(ii)))  ! Calc of excess solute for indiv. particle
  xsum=xsum+(m(ii)−(phi*v(ii)))  ! Summing excess variance for all particles
  end do
x2=(m(ii)-(phi*v(ii)))**2  ! Calc of variance for indiv. particle
x2sum=x2sum+(m(ii)-(phi*v(ii)))**2  ! Summing variance for all particles

end do

if(loop.eq.0) then
  average_kernel=0.0
end if

!***** Calculation of Monte Carlo coagulation and binder flow event rates *****
! coag_rate=(C0*V_R)*(average_kernel*(conc**2.0)/2.0)
! binder_rate=(C0*V_R)*tau_ratio/n_binder1
! rate_ratio=coag_rate/binder_rate

!***** Write statistical parameters to screen *****
! write(*,*), "x43 =", x43
! write(*,*), "gamma =", gamma
! write(*,*), "C0 =", C0
! write(*,*), "V_R =", V_R
! write(*,*), "average_kernel =", average_kernel
! write(*,*), "conc =", conc
! write(*,*), "n_binder1 =", n_binder1
! write(*,*), "coag_rate =", coag_rate
! write(*,*), "binder_rate =", binder_rate
! write(*,*), "rate_ratio =", rate_ratio

!***** Write statistical parameters to output files *****
! open (11, file = 'excess_solute.txt')  ! For storage of excess solute data
! open (21, file = 'segregation.txt')   ! For storage of segregation index data
! write (*,*), "x2sum =", x2sum  ! Writing indiv. particle variance to screen (for troubleshooting)
! write (11,*), vav, "", "", xsum/n
! write (21,*), vav, "", "", x2sum/n, "", (x2sum/n)/(vav**2), "", "", x2sum/mass  ! Writing indiv. particle variance to file segregation.txt
! write (41,*), vav, "", "", time, "", "", (x2sum/n)/(vav**2)
write (81,*) vav, "", conc, "", time, "", average_eta, "", average_eta_ap, "", &
   (tauc_tauhevap*LOD_dry*average_ap*((1.0*d0-vb)*cw_sat - moisture)*
    conc)
write (111,*) wetavg, "", time, "", dryavg, "", LOD_wet, "", "", LOD_dry
!!!write (121,*) dried_gran_vavg, "", time, "", "", dried_d43, "", "", rate_ratio
write (121,*) dryavg, "", time, "", "", dried_d43, "", "", rate_ratio
write (151,*) conc,"", "", mass_conc,"", "", moisture,"", "", time
cB = binder/(1.0*n)*conc
cP = part/(1.0*n)*conc
cBcP = conc-cB
write (91,*) cB, "", "", cP, "", "", conc

***** Write more statistical parameters to screen *****
!!!write(*,*) "mass =", mass
!!!write(*,*) "solute=" , solute
!!!write(*,*) "n =", n
!!!write(*,*) "phi =", phi
!!!write(*,*) "phi0 =", phi0
!!!write(*,*) "vav =", vav
!!!write(*,*) "time =", time
!!!write(*,*) "average kernel =", average_kernel
!!!write(*,*) "binder =", binder
!!!write(*,*) "part =", part
!!!write(*,*) "bs_ratio1 =", bs_ratio1
!!!write(*,*) "frc1 =", frc1
!!!write(*,*) "eta1 =", eta1
!!!write(*,*) "bs_ratio2 =", bs_ratio2
!!!write(*,*) "frc2 =", frc2
!!!write(*,*) "eta2 =", eta2
!!!write(*,*) "Psi =", Psi
!!!write(*,*)
end subroutine stats

Appendix–A/stats.f95
A.13 Random Number Generator

FUNCTION ran0(idum)
INTEGER idum, IA, IM, IQ, IR, MASK
REAL ran0, AM
PARAMETER (IA=16807, IM=2147483647, AM=1./IM, &
IQ=127773, IR=2836, MASK=123459876)

"Minimal" random number generator of Park and Miller. Returns a
uniform random deviate between 0.0 and 1.0. Set or reset idum
to
any integer value (except the unlikely value MASK) to
initialize
the sequence; idum must not be altered between calls for
successive
deviates in a sequence.
INTEGER k
idum = ieor(idum, MASK)  ! XORing with MASK allows use of zero and
k = idum / IQ          ! other simple bit patterns for idum.
idum = IA * (idum - k * IQ) - IR * k  ! Compute idum=mod(IA*idum, IM)
without
if (idum < 0) idum = idum + IM  ! overflows by Schrage's method.
ran0 = AM * idum     ! Convert idum to a floating result.
idum = ieor(idum, MASK)  ! Unmask before return.
return
END

! (C) Copr. 1986–92 Numerical Recipes Software, #.
A.14 Optional Subroutine to Evaluate Particle Array Before and After Coagulation

```plaintext
subroutine wrt_before(i, j, k)
use particle
integer ii
!
write particle array BEFORE coagulation
write(*,*), 'BEFORE'
write(*,*), "particles picked to coagulate: i=", i, "j=", j
write(*,*), "particle picked to be added: k=", k

do ii=1, n
if(ii .eq. i .or. ii .eq. j) then
  write(*,*), '*', v(ii), m(ii), w(ii)
else if (ii .eq. k) then
  write(*,*), 'r', v(ii), m(ii), w(ii)
else
  write(*,*), '', v(ii), m(ii), w(ii)
end if
end do
return
end

subroutine wrt_after(i, j, k)
use particle
integer ii
!
write particle array AFTER
write(*,*), 'AFTER' ! begin write array
write(*,*), "new particle saved in position i=", i
write(*,*), "particle k=", k, "saved in position j=", j

do ii=1, n
if(ii .eq. i .or. ii .eq. j) then
  write(*,*), '*', v(ii), m(ii), w(ii)
else if (ii .eq. k) then
  write(*,*), 'r', v(ii), m(ii), w(ii)
else
  write(*,*), '', v(ii), m(ii), w(ii)
```

end if
end do !end write array
end

Appendix–A/optional_subs.f95
Appendix B

Input files for Continuous Addition of Binder

B.1 Input files overview

While several parameters must be defined to start the simulation, only three minor input files are utilized in this code. From these files, the coagulation constants ($K_0$) for particle-particle and particle-binder interactions and the desired sieve cut widths for analyzing the final particle distribution are given. These files are discussed in the following sections.
B.2 Initial Dist Input1.txt

This input file is entitled “Initial Dist Input1” and simply contains four rows of text. The first two lines correspond to the particle-particle and particle-binder droplet $K_0$ values, respectively. As per the discussion in Chapter 4, the particle-particle $K_0$ value is to always be 100 times greater than the particle-binder $K_0$. The last two lines in the file correspond to the size and number of powder particles for powder 1 (of 2). Each line contains two numbers separated by a comma. The number on the left corresponds to the granule size and the right-hand number corresponds to the number of granules of that size. For monodisperse conditions, lines three and four should be identical. An example of the text file is given below:

```
9.00d-14
9.00d-12
10.0,3599.0
10.0,3599.0
```

While this example details a monodisperse case, other distributions such as a normal or log-normal distribution may be utilized here.
B.3 Initial Dist Input2.txt

This input file, “Initial Dist Input2.txt”, contains only two rows of text and corresponds to the size and number of powder particles for the second of two powders. As before, each line contains two numbers separated by a comma. The number on the left corresponds to the granule size and the right-hand number corresponds to the number of granules of that size. Once again, these lines should be identical for monodisperse conditions. An example of the text file is given below:

26.0,1401.0
26.0,1401.0

While this example details a monodisperse case, other distributions such as a normal or log-normal distribution may be utilized here.
B.4 Final Dist Sieves.txt

“Final Dist Sieves.txt”, is used when it is desired to evaluate sieve cuts of the final particle population. The file contains only one column of text that specifies the upper value for each sieve cut. An example of the text file is:

9.0
11.0
13.0
15.0
18.0
22.0
26.0
31.0
37.0
43.0
50.0
60.0
75.0
90.0
105.0
125.0
150.0
180.0
210.0
250.0
300.0
360.0
430.0
510.0
610.0
730.0
870.0
1030.0
1230.0
1470.0
1750.0

The values chosen for this file are completely arbitrary.
Bibliography


binder: effects of bed temperature and droplet size,” in Proceedings of World Congress of Particle Technology 3, p. paper 103.


Vita
Carl L. Marshall, Jr.

Education
M.S. Chemical Engineering, Bucknell University, 2002.

Employment
Senior Process Engineer, Merck & Co., Inc., December 2011 - Present.
Organic Chemistry intern, PA Department of Environmental Protection, Summer 1998
Organic Chemistry intern, PA Department of Environmental Protection, Summer 1997

Selected Honors and Awards
Recipient of the Penn State Department of Chemical Engineering’s Personal Excellence Award for Outstanding Teaching Assistant: 2010
Merck Award for Excellence for role in commercialization product transfer: 2007
Tau Beta Pi Engineering Honor Society induction: November 1999
Phi Kappa Phi Honor Society induction: April 1998
Phi Eta Sigma Freshman Honor Society induction: 1995

Publications