Stochastic Solution of
Multi-Dimensional Population Balances

Michael J Goodson

Trinity College

A dissertation submitted for the degree of Doctor of Philosophy at the
University of Cambridge

September 2007
Preface

The work presented in this dissertation was undertaken at the Department of Chemical Engineering, University of Cambridge, between October 2000 and August 2007. This dissertation is the result of my own work and includes nothing which is the outcome of work done in collaboration except where specifically indicated in the text. Neither the present dissertation, nor any part thereof, has been submitted previously for a degree to this or any other university. This thesis contains approximately 30000 words and 40 figures in total. The author was funded by the EPSRC.

Some of the work in this dissertation has been published:


Mike Goodson
September 2007.
Summary

This work examines the stochastic simulation approach to solving the multi-dimensional population balance equation, and demonstrates the efficiency and ease of application to a number of different areas.

Initially, existing stochastic solution methods for the Smoluchowski coagulation equation are examined. The Smoluchowski equation is extended to include a source term and the performance of an improved direct simulation algorithm is compared with existing methods. Application of this method, in conjunction with the method of moments, to nanoparticle synthesis in flames demonstrates the ability to resolve particle size distributions in the case where multiple processes are occurring.

Further extension, to include breakage, is formulated, and two algorithms are investigated: the Direct Simulation Algorithm (DSA) and the Mass Flow Algorithm (MFA). Both algorithms are very efficient, when coupled with efficient storage of functions of particle size distributions and a constant number simulation technique. DSA proves more effective for simulating lower (less than one) moments of the size distribution, while MFA is very efficient for higher moments.

The first multi-dimensional case studied is that of a rotating disc contactor with mass transfer occurring. Extension of the stochastic simulation method to multiple dimensions is intuitive and efficient. Comparison with existing numerical simulations shows that the stochastic method can reproduce these results.

Finally, a tentative approach to a new model for wet powder agglomeration is proposed. A three-dimensional model for aggregate structure is suggested, with initial approaches to solve some of the modelling problems associated with agglomeration behaviour. The stochastic simulation method proves ideal for implementing these models, and, coupled with the results of well-chosen experiments, shows great promise in the development and validation of these models.
I would like to thank my supervisor, Dr Markus Kraft, without whose advice, guidance and encouragement the work presented in this dissertation would not have been possible.

Most of this work has been done sat at a computer, so I am indebted to those people with whose experimental work I have been able to compare my results. These are Dr Dirk Grosschmidt of Universität Karlsruhe, Dr Menwer Attarkih and Dr Martin Simon of Technische Universität Kaiserslautern, and Dr Sarah Forrest of the University of Cambridge.

I would also like to extend my thanks to all the members of the Computational Modelling Group here in Cambridge. Through academic insight, technical discussions and IT expertise, they have helped me find my way through many tricky problems.

Trinity College have offered me a great deal of support, including generous financial backing for overseas conferences.

Of course, along with the hard-working days, there have been many happy nights. My thanks go to everyone who I have been for a drink (or two) with, whether from the Chemical Engineering Department, or from those great social centres of Portugal Street and St Barnabas Road. There are too many names to fit on this page, so I won’t disappoint the many by mentioning a few.

And finally, I would like to thank my parents, who have given me unconditional love and support in everything I do.
# Contents

Preface

Summary

Acknowledgements

List of Figures

List of Tables

List of Symbols

1 Introduction 1

1.1 Population Balance Modelling 1

1.2 Solution Methods 3

1.3 Stochastic Simulation 4

1.4 Multi-dimensional Population Balance Models 6

1.5 Motivation 7

1.6 Aims 8

2 The Smoluchowski Coagulation Equation 10

2.1 Existing Stochastic Algorithms 10

2.2 The Stochastic Model 11

2.2.1 Markov Process With Fictitious Jumps 11

2.2.2 Organisation Of Particle System 14

2.2.3 Majorant Kernel 15

Appendices 19

A Justification Of Choice Of Majorant Kernel 19
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.7</td>
<td>Numerical Results</td>
<td>80</td>
</tr>
<tr>
<td>6.7.1</td>
<td>Convergence Properties</td>
<td>82</td>
</tr>
<tr>
<td>6.7.2</td>
<td>DSA or MFA?</td>
<td>87</td>
</tr>
<tr>
<td>6.8</td>
<td>Minimum Particle Size</td>
<td>87</td>
</tr>
<tr>
<td>6.8.1</td>
<td>Case 1</td>
<td>91</td>
</tr>
<tr>
<td>6.8.2</td>
<td>Case 2</td>
<td>92</td>
</tr>
<tr>
<td>6.9</td>
<td>Ternary Breakage</td>
<td>94</td>
</tr>
<tr>
<td>Appendices</td>
<td></td>
<td>99</td>
</tr>
<tr>
<td>A</td>
<td>Solution of Test Case</td>
<td>99</td>
</tr>
<tr>
<td>B</td>
<td>Algebra relating to ternary breakage</td>
<td>101</td>
</tr>
<tr>
<td>7</td>
<td>Simulation of a 2-Component RDC</td>
<td>104</td>
</tr>
<tr>
<td>7.1</td>
<td>Modelling Coalescence and Breakage</td>
<td>104</td>
</tr>
<tr>
<td>7.2</td>
<td>Coalescence and Breakage Models</td>
<td>105</td>
</tr>
<tr>
<td>7.2.1</td>
<td>Turbulent Dispersions</td>
<td>105</td>
</tr>
<tr>
<td>7.2.2</td>
<td>Oldshue-Rushton Contactor</td>
<td>108</td>
</tr>
<tr>
<td>7.2.3</td>
<td>Oscillatory Baffled Reactor</td>
<td>108</td>
</tr>
<tr>
<td>7.2.4</td>
<td>Bubble Column Reactor</td>
<td>109</td>
</tr>
<tr>
<td>7.2.5</td>
<td>Rotating Disc Contactor</td>
<td>110</td>
</tr>
<tr>
<td>7.2.6</td>
<td>Kühni Column</td>
<td>111</td>
</tr>
<tr>
<td>7.2.7</td>
<td>Pulsed Sieve-Plate Column</td>
<td>111</td>
</tr>
<tr>
<td>7.2.8</td>
<td>Packed Liquid Extraction Column</td>
<td>112</td>
</tr>
<tr>
<td>7.2.9</td>
<td>Remarks</td>
<td>113</td>
</tr>
<tr>
<td>7.3</td>
<td>Experimental Comparison</td>
<td>114</td>
</tr>
<tr>
<td>7.4</td>
<td>Models</td>
<td>114</td>
</tr>
<tr>
<td>7.4.1</td>
<td>Rise Velocity</td>
<td>114</td>
</tr>
<tr>
<td>7.4.2</td>
<td>Coalescence</td>
<td>115</td>
</tr>
<tr>
<td>7.4.3</td>
<td>Breakage</td>
<td>117</td>
</tr>
<tr>
<td>7.5</td>
<td>Simulation Results</td>
<td>118</td>
</tr>
<tr>
<td>7.6</td>
<td>Remarks</td>
<td>121</td>
</tr>
<tr>
<td>8</td>
<td>Simulation of a 3-Component RDC</td>
<td>122</td>
</tr>
<tr>
<td>8.1</td>
<td>Mass Transfer to/from a Sphere</td>
<td>122</td>
</tr>
<tr>
<td>8.1.1</td>
<td>Mass Transfer Resistance In The Continuous Phase</td>
<td>122</td>
</tr>
<tr>
<td>8.1.2</td>
<td>Mass Transfer Resistance In The Discrete Phase</td>
<td>123</td>
</tr>
</tbody>
</table>
CONTENTS

8.1.3 Mass Transfer Resistance In Both Phases .................. 125
8.1.4 Which Phase? .............................................. 125
8.2 Multi-dimensional Stochastic Simulation ...................... 126
  8.2.1 Stochastic Particle Array ................................ 126
8.3 Stochastic Treatment of Mass Transfer ....................... 128
  8.3.1 Continuous Treatment ...................................... 129
  8.3.2 Discrete Treatment ......................................... 130
  8.3.3 Process Deferment ......................................... 132
8.4 Algorithm Testing .............................................. 133
8.5 Simulation of mass transfer in a rotating disc contactor .... 136

9 Further Work .................................................. 140
  9.1 Scope of the Chapter .......................................... 140
  9.2 Wet Powder Agglomeration .................................... 141
    9.2.1 Multi-Dimensional Population Balance Model ............ 141
    9.2.2 Model Extension .......................................... 144
    9.2.3 Relating Surface Area to Pore Volume .................... 145
    9.2.4 Collision Softness ....................................... 147
    9.2.5 Model Implementation .................................... 148

10 Conclusions .................................................. 151

Bibliography ...................................................... 154
List of Figures

3.1 Moments of the particle size distribution for case 1 (inflow only). 31
3.2 Moments of the particle size distribution for case 2 (outflow only). 32
3.3 Moments of the particle size distribution for case 3 (inflow and outflow). 33
3.4 Moments of the particle size distribution for case 4 (inflow and outflow with $N \neq n_{in}$). 34

4.1 Upper bound (dashed lines) and lower bound (solid lines) of the particle size distributions at various times. 42
4.2 Upper bound (dashed lines) and lower bound (solid lines) of the moments of the particle size distribution. 43
4.3 Order of convergence for improved stochastic algorithm. The solid line shows the slope $\frac{1}{N}$. 46
4.4 CPU time for a single run for the Gillespie algorithm (solid) and our improved algorithm (dotted). 48
4.5 CPU time for a single run for the linear majorant and for our new majorant. $t_{sim}$ is the simulation time. 49
4.6 Comparison of the efficiency of the two majorants. 50
4.7 CPU time for a single run for the linear majorant and our new majorant. $N$ is the particle number. 50
4.8 Order of convergence for the Gillespie algorithm. The solid line shows the slope $\frac{1}{N}$. 54

5.1 Molefraction profiles of major gas-phase species and production rate of SiO$_2$ (Grosschmidt et al., 2003). 58
5.2 Mean particle mass as a function of the height above the burner for the method of moments and stochastic simulation. The discrepancy between simulation and measurements is explained by the fact that the ionization of the particles in the experiments are not accounted for in the models. ........................................ 59

5.3 Comparison of the particle size distribution obtained from stochastic simulation (solid lines) with an assumed shape lognormal probability density function (dashed line). The parameters for the assumed shape probability density function were obtained from the expectation and variance of the distribution as $A = 5.818$, $B = 0.996$. 61

5.4 Comparison of stochastic simulation (solid lines) and assumed shape lognormal fit (dashed lines) for different heights above the burner 62

6.1 A binary tree, of height 3. ........................................ 74

6.2 Single run CPU time comparison for the bins and binary methods. 74

6.3 Schematic representation of a particle duplication step following a coagulation event. ........................................ 76

6.4 Schematic representation of a combined coagulation event and particle duplication step. ........................................ 76

6.5 Schematic representation of a combined breakage event and particle removal step. ........................................ 77

6.6 Comparison of the zeroth moment predicted by Direct Simulation (DSA) and Mass Flow (MFA). ........................................ 82

6.7 Comparison of the fifth moment predicted by Direct Simulation (DSA) and Mass Flow (MFA). ........................................ 83

6.8 Convergence to the analytical solution of the 2nd to 5th moments predicted by the mass flow algorithm. The solid lines show slopes of -1. In this study, $n \times L$ is kept constant at 104857600. ........ 84

6.9 Convergence to the analytical solution of the zeroth moment predicted by the mass flow algorithm. The solid line shows a slope of -1/3. In this study, $n \times L$ is kept constant at 104857600. ........ 85
6.10 Convergence to the analytical solution of the zeroth moment predicted by the mass flow algorithm. In this study, $n \times L$ is kept constant at 104857600. The number of repeated runs is sufficient to smooth out any fluctuations in the confidence intervals, but the systematic error is still significant. 

6.11 Convergence to the analytical solution of the 0th and 2nd to 5th moments predicted by the direct simulation algorithm. The solid lines show slopes of -1. In this study, $n \times L$ is kept constant at 104857600. 

6.12 Single run CPU time required to predict the zeroth moment with a given systematic error. 

6.13 Single run CPU time required to predict the second moment with a given systematic error. 

6.14 Single run CPU time required to predict the fifth moment with a given systematic error. 

6.15 Behaviour of the Mass Flow Algorithm with the imposition of a minimum particle size, $x_{\text{min}}$. The confidence bounds are for the solutions given by MFA. The solid lines are the converged solutions given by DSA. 

6.16 Predicted daughter droplet size distributions according to Equation (6.45). 

7.1 Comparison of simulated and experimental results for a butylacetate-water system in a laboratory scale rotating disc contactor. Conditions: continuous phase flowrate = 100 l/min; discrete phase flowrate = 100 l/min. Dotted line = simulated inlet distribution; Solid lines = upper and lower bounds of simulated outlet distribution; $\bullet$ = measured inlet distribution; $\bigtriangleup$ = measured outlet distribution.
7.2 Comparison of simulated and experimental results for a butylacetate-water system in a laboratory scale rotating disc contactor. Conditions: continuous phase flowrate = 50 l/min; discrete phase flowrate = 100 l/min. Dotted line = simulated inlet distribution; Solid lines = upper and lower bounds of simulated outlet distribution; • = measured inlet distribution; ▲ = measured outlet distribution. 120

8.1 Test case simulation results for three different algorithms. 135

8.2 Comparison of simulated results (dashed lines) and assumed shape normal distribution (solid). The parameters for the normal distribution are calculated by using the analytical solution for the mean and the method of least squares to determine the standard deviation. 137

8.3 Convergence of the standard deviation with ∆Y at various simulation times. The solid lines have a slope of 0.5. 138

8.4 Simulation of mass transfer in a rotating disc contactor. Comparison of simulated results (dotted lines) with the results of Attarakih et al. (2006) (solid lines). 139

9.1 Schematic of an aggregate particle. The particle occupies a total volume (including pore spaces) of \( v = \frac{4}{3} \pi r^3 \). 146

9.2 Illustration of simulated aggregate properties for different values of rotational speed (corresponding to different values of \( \varepsilon_{\text{crit}} \)). Solid lines show \( \varepsilon_{\text{crit}} = 0.1 \); dashed lines show \( \varepsilon_{\text{crit}} = 0.15 \). 150
List of Tables

4.1 Computational study for improved stochastic algorithm \((N \times L = 6.4 \times 10^7)\). ................................................................. 45
4.2 Computational study for Gillespie algorithm \((N \times L = 8 \times 10^5)\). 47
8.1 Comparison of run times for three stochastic algorithms. ....... 134
8.2 Fitted normal distribution for the discrete simulation algorithm. 136
9.1 Probability of coalescence as function of properties of granules \(i\) and \(j\). ................................................................. 142
9.2 Probability of coalescence as function of properties of granules 1 and 2. The particles are described as either big or small (b/s), dry or wet (d/w) and soft or hard (s/h). ............................................. 144
List of Symbols

**Roman symbols**

- $a$  
  surface area, [m²].
- $a_\alpha$  
  error bound associated with confidence level $\alpha$, [-].
- $A$  
  cross-sectional area, [m²].
- $\lambda$  
  parameter in the log-normal distribution, [-].
- $A^*$  
  stochastic operator, [-].
- $b_1 \rightarrow b_5$  
  constants used in breakage models, [variable].
- $b_z$  
  upper size bound of bin $z$, [-].
- $B$  
  parameter in the log-normal distribution, [-].
- $B(m : n)$  
  complete Beta function $= B(1; m, n)$, [-].
- $B(x; m : n)$  
  incomplete Beta function $= \int_0^x y^{m-1} (1 - y)^{n-1} dy$, [-].
- $c(x, t)$  
  number per unit volume of particles of size $x$ at time $t$  
  (in the discrete form of the population balance equation, [m⁻³]).
- $c(x, t)dx$  
  number per unit volume of particles in the size range $[x \rightarrow x + dx]$ at time $t$  
  (in the continuous form of the population balance equation, [m⁻³]).
- $c_1 \rightarrow c_5$  
  constants used in coalescence models, [variable].
- $c_{IP}$  
  instability constant, [-].
- $c_p$  
  probabilistic upper bound for statistical error, [variable].
- $c_{stat}$  
  approximation of $e_{stat}$, [variable].
- $c_{tot}$  
  approximation of $e_{sys}$, [variable].
- $C_0$  
  summation used in Gillespie stochastic algorithm, [m³ s⁻¹].
- $C_i$  
  summations used in Gillespie stochastic algorithm, [m³ s⁻¹].
- $d$  
  particle diameter, [m].
<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$d_{12}$</td>
<td>harmonic mean of two diameters $d_1$ and $d_2$, [m].</td>
</tr>
<tr>
<td>$d_{32}$</td>
<td>Sauter mean diameter, [m].</td>
</tr>
<tr>
<td>$d_e$</td>
<td>eddy diameter, [m].</td>
</tr>
<tr>
<td>$D$</td>
<td>diffusivity, [m$^2$ s$^{-1}$].</td>
</tr>
<tr>
<td>$D_c$</td>
<td>cell diameter, [m].</td>
</tr>
<tr>
<td>$D_F$</td>
<td>fractal dimension, [-].</td>
</tr>
<tr>
<td>$D_r$</td>
<td>rotor diameter, [m].</td>
</tr>
<tr>
<td>$D_s$</td>
<td>stator diameter, [m].</td>
</tr>
<tr>
<td>$D_S$</td>
<td>surface area fractal dimension, [-].</td>
</tr>
<tr>
<td>$e$</td>
<td>error in predicted functional, [variable].</td>
</tr>
<tr>
<td>$f(x, y)$</td>
<td>rate at which a particle of size $x$ and a particle of size $y$ are formed from breakage of a particle of size $x + y$, [s$^{-1}$].</td>
</tr>
<tr>
<td>$f_{i,j}$</td>
<td>coagulation probability for collision between aggregates $i$ and $j$, [-].</td>
</tr>
<tr>
<td>$f_X(x)$</td>
<td>probability density function, [m$^{-3}$].</td>
</tr>
<tr>
<td>$f_y$</td>
<td>a ratio of two particle sizes $= \frac{y}{x}$, [-].</td>
</tr>
<tr>
<td>$F(t)$</td>
<td>functional of the particle size distribution (e.g. moment), [variable].</td>
</tr>
<tr>
<td>$Fo$</td>
<td>Fourier number $= \frac{4Dd}{g}$, [-].</td>
</tr>
<tr>
<td>$g$</td>
<td>acceleration due to gravity $= 9.81$ m s$^{-2}$, [m s$^{-2}$].</td>
</tr>
<tr>
<td>$g(x)$</td>
<td>breakage frequency of a particle of size $x$, [s$^{-1}$].</td>
</tr>
<tr>
<td>$h(d_1, d_2)$</td>
<td>collision frequency of particles of size $d_1$ and $d_2$, [m$^3$ s$^{-1}$].</td>
</tr>
<tr>
<td>$h_c$</td>
<td>height of a compartment in a rotating disc contactor, [m].</td>
</tr>
<tr>
<td>$I(t)$</td>
<td>rate of particle inception, [s$^{-1}$].</td>
</tr>
<tr>
<td>$J$</td>
<td>stochastic jump process, [m$^{-6}$].</td>
</tr>
<tr>
<td>$k$</td>
<td>individual mass transfer coefficient, [m s$^{-1}$].</td>
</tr>
<tr>
<td>$k_o$</td>
<td>overall mass transfer coefficient, [m s$^{-1}$].</td>
</tr>
<tr>
<td>$k_B$</td>
<td>Boltzmann’s constant $= 1.38 \times 10^{-23}$ J K$^{-1}$, [J K$^{-1}$].</td>
</tr>
<tr>
<td>$k_v$</td>
<td>slowing factor, [m s$^{-1}$].</td>
</tr>
<tr>
<td>$K(x)$</td>
<td>general rate term in population balance equation, [s$^{-1}$].</td>
</tr>
<tr>
<td>$K(x, y)$</td>
<td>coagulation rate kernel, [m$^3$ s$^{-1}$].</td>
</tr>
<tr>
<td>$\hat{K}(x, y)$</td>
<td>coagulation majorant kernel, [m$^3$ s$^{-1}$].</td>
</tr>
<tr>
<td>$Kn$</td>
<td>Knudsen number $= \frac{2k_B}{\lambda d}$, [-].</td>
</tr>
<tr>
<td>$l$</td>
<td>counter for number of repetitions, [-].</td>
</tr>
<tr>
<td>$L$</td>
<td>total number of repetitions, [-].</td>
</tr>
<tr>
<td>$m$</td>
<td>mass, [kg].</td>
</tr>
</tbody>
</table>
LIST OF SYMBOLS

\( m_1 \)  monomer mass, [kg].

\( m_i(t) \)  \( i^{th} \) moment of the particle size distribution, [variable].

\( m_{RZ} \)  exponent in Richardson-Zaki correlation, [-].

\( M \)  number of discretised time intervals, [-].

\( Mo \)  Morton number = \( \frac{g \mu}{\rho_c^4 (\rho_c - \rho_d)} \), [-].

\( n \)  number of stochastic particles, [-].

\( N \)  drum revolutions, [rev].

\( N \)  normalisation parameter for stochastic particle ensemble (analogous to sample volume), [m^3].

\( N' \)  updated value of \( N \) after particle duplication/removal and renormalisation, [m^3].

\( p \)  breakage probability, [-].

\( p \)  parameter in the Beta distribution, [-].

\( p(x) \)  representation of a stochastic particle ensemble, \( i.e., \) a realisation of \( P^N(x,t), [m^{-6}] \).

\( P^N(x,t) \)  a sequence of stochastic jump processes, \( i.e., \) a measure-valued solution to the population balance equation, [m^{-6}].

\( Pe \)  Peclet number (see Page 126 for definition), [-].

\( q(x) \)  representation of a stochastic particle ensemble, \( i.e., \) a realisation of \( Q^N(x,t), [m^{-3}] \).

\( Q \)  flow rate, [m^3 s^{-1}].

\( Q^N(x,t) \)  a sequence of stochastic jump processes, \( i.e., \) a measure-valued solution to the population balance equation in the mass flow case, [m^{-3}].

\( r \)  particle radius, [m].

\( Re \)  Reynolds number = \( \frac{\rho_c v_d \mu_c}{\rho_c} \), [-].

\( s \)  dummy variable used to calculate waiting time, \( \tau, [s] \).

\( Sc \)  Schmidt number = \( \frac{\mu_c}{\rho_c D_d} \), [-].

\( Sh \)  Sherwood number = \( k c \), [-].

\( St_{def} \)  Stokes deformation number = \( \frac{\rho_c v_d^2}{2 \gamma} \), [-].

\( t \)  time, [s].

\( t_i \)  inception time, [s].

\( t_k \)  characteristic time for mass transfer, [s].

\( t_{sr} \)  single run CPU time, [s].
LIST OF SYMBOLS

$t_{tot}$ total time, [s].

$T$ temperature, [K].

$u$ velocity, [m s$^{-1}$].

$u_T$ droplet terminal velocity, [m s$^{-1}$].

$u_r$ slip velocity between two phases, [m s$^{-1}$].

$v$ particle volume, [m$^3$].

$V$ compartment volume, [m$^3$].

$We$ Weber number = $\frac{\rho u^2 d}{\sigma}$, but see Chapter 7 for specific forms used in this work, [-].

$x$ particle size, which is conserved on coalescence; either number of monomer units in the discrete case, [-], or, in the continuous case, volume, [m$^3$], or mass [kg].

$y$ particle size, which is conserved on coalescence; either number of monomer units in the discrete case, [-], or, in the continuous case, volume, [m$^3$], or mass [kg].

$Y$ mass fraction, [-].

$z(x, y)$ function used in the appendices to Chapter 2, [-].

Greek symbols

$\alpha$ confidence level, [-].

$\alpha$ index of stochastic event, [-].

$\alpha_1$ scaling factor in coagulation kernel, [-].

$\alpha_2$ scaling factor in coagulation kernel, [-].

$\alpha_z$ number of particles in bin $z$, [-].

$\beta$ size ratio defining bin sizes, [-].

$\beta_\nu(x, y) dy$ probability that the size of a particle formed on breakage of a particle of size $x$ into $\nu$ daughter particles is in the size range $[y \rightarrow y + dy]$, [-].

$\gamma$ degree of homogeneity, [-].

$\Gamma(m)$ gamma function = $\int_0^\infty x^{m-1}e^{-x}dx$, [-].

$\varepsilon$ exponent in size distribution function, [-].

$\varepsilon$ porosity, [-].

$\varepsilon$ power input per unit mass, [m$^2$ s$^{-3}$].
LIST OF SYMBOLS

\( \zeta(t) \) approximation of \( F(t) \), [variable].
\( \eta_1 \) empirical mean value of \( \Xi \), [variable].
\( \eta_2 \) empirical variance of \( \Xi \), [variable].
\( \kappa \) distribution coefficient, [-].
\( \lambda \) a constant, [-].
\( \lambda_b \) breakage efficiency, [-].
\( \lambda_c \) coalescence efficiency, [-].
\( \lambda_f \) mean free path, [m].
\( \mu \) viscosity, [kg m\(^{-1}\) s\(^{-1}\)].
\( \nu \) number of daughter droplets formed on breakage, [-].
\( \xi \) collision softness, [-].
\( \xi_\text{realisation} \) realisation of random variable \( \Xi \), [variable].
\( \Xi \) a random variable, approximating \( F(t) \), [variable].
\( \rho \) density, [kg m\(^{-3}\)].
\( \rho(p) \) waiting time parameter for stochastic process, [s\(^{-1}\)].
\( \sigma \) surface tension, [N m\(^{-1}\)].
\( \sigma_y \) yield stress, [N m\(^{-2}\)].
\( \tau \) waiting time between stochastic events, [s].
\( \tau_{\text{res}} \) mean residence time for inflow/outflow, [s].
\( \phi \) holdup (volume fraction of discrete phase), [-].
\( \phi(x) \) function of particle size used to calculate \( F(t) \), [variable].
\( \Phi(p) \) functional of stochastic particle ensemble, used to approximate
\( F(t) \), [variable].
\( \psi \) number of bins (groups) of particles, [-].
\( \omega \) rotational velocity, [rev s\(^{-1}\)].

**Subscripts**

0 initial, i.e., at \( t = 0 \).
a air.
c continuous phase.
crit critical.
d discrete phase.
e equilibrium.
LIST OF SYMBOLS

\textit{eff}  
\textit{g}  
\textit{i}  
\textit{in}  
\textit{j}  
\textit{k}  
\textit{K}  
\textit{l}  
\textit{min}  
\textit{max}  
\textit{out}  
\textit{p}  
\textit{s}  
\textit{sys}  
\textit{stat}  
\textit{z}  

effective.
breakage.
particle index, \textit{i.e.}, position in stochastic array.
inflowing.
particle index, \textit{i.e.}, position in stochastic array.
mass transfer.
coagulation.
liquid.
particle index, \textit{i.e.}, position in stochastic array.
minimum.
maximum.
outflowing.
pore.
solid.
systematic.
statistical (\textit{i.e.}, due to fluctuations).
bin index.
Chapter 1

Introduction

“Granule porosity, binder content, and composition are all independent granule properties that can significantly affect granule growth behaviour and final product properties. Traditional one-dimensional population balances, which assume that granule size is the only independent granule property controlling coalescence, are unable to account for these effects.”


Although the above quote refers specifically to the field of wet powder agglomeration (see Chapter 9), there are several fields in which multi-dimensional population balances are becoming recognised as a necessary tool in modelling.

1.1 Population Balance Modelling

Population balance modelling has a long history. The field of statistical mechanics (relating macroscopic properties of a population to the microscopic properties of its component parts) has its origins with the Boltzmann equation (Boltzmann, 1872). An early example of a population balance equation is Smoluchowski’s equation to describe the pure coagulation\(^1\) case (von Smoluchowski, 1916). A

\(^{1}\)In the literature, many terms are used to describe the processes of coalescence (coagulation, aggregation, agglomeration) and breakage (breakup, fragmentation). Occasionally, separate terms are used in the literature to describe subtly different processes (such as coagulation with rearrangement versus two particles merely sticking together). In this work, to avoid confusion,
statistical mechanical population balance description was constructed by Hulbert and Katz (1964), while the application to particulate processes was specified by Randolph and Larson (1988). The applications of population balance modelling are widespread; any system of interacting particles (molecules, droplets, agglomerates or any other discrete entities) that are characterised by one or more internal co-ordinates can be modelled using a population balance approach.

One early application of population balance modelling was in liquid-liquid systems. Valenta et al. (1966) modelled breakage and Valenta and Amundson (1966) modelled both breakage and coalescence. A standard reference for a phenomenological approach to modelling coalescence and breakage in a turbulent dispersion remains the work of Coulaloglou and Tavlarides (1976, 1977). This work has been updated many times (e.g. Bapat et al., 1983; Tsouris and Tavlarides, 1994) and built on by other researchers (e.g. Sovova, 1981; Luo and Svendsen, 1996), but many of the concepts remain the same.

The growth of droplets in clouds by discrete interactions with other drops (rather than continuous accretion of mass) was a phenomenon first recognised by Telford (1955). The stochastic nature of such interactions prompted some of the first stochastic algorithms proposed for solution of the population balances (Gillespie, 1972, 1974, 1975)

The formation of solid nano-particles from gas-phase species is one of the few areas of application where the rate laws for particle interaction seem to be well established and accepted. Coalescence rate kernels have been proposed for the continuum, transition, and free-molecular regimes (Friedlander, 1977) based on the kinetic theory of gases. These have been used largely unchanged for applications in the formation of soot (Frenklach and Wang, 1994; Balthasar, 1999), silica (Lindackers et al., 1997), iron oxide (Janzen and Roth, 2001) and titania (Spicer et al., 2002) to name but a few. Present work in this area now tends to involve coupling population balance modelling to computational fluid dynamics (Mühlenweg et al., 2002; Grosschmidt et al., 2006)

In crystallisation problems, aggregation and breakage of particles is accompanied by a continuous growth process as material comes out of solution. Studies on batch and continuous systems, and specifically on problems involving gelation,
have been carried out by Smit et al. (1994a, b, 1995)

Wet powder agglomeration problems have long been attacked with population balance approaches (Sastry and Fuerstenau, 1973; Adetayo and Ennis, 1997, 2000; Liu and Litster, 2002), but these models have been mainly empirical.

Further systems where population balances have been applied include flocculation in water treatment processes (Ducoste, 2002), granulation and pelletisation (Ouchiyama and Tanaka, 1982), polymerisation and depolymerisation (Stockmayer, 1943; Ziff and McGrady, 1985).

A very thorough study of the field of population balances and their applications to particulate systems has been made by Ramkrishna (2000).

### 1.2 Solution Methods

The population balance equation (pbe) gives, when solved, a complete statistical description of the system, \textit{i.e.}, a number density function for a set of particles in terms of size, concentration or other internal co-ordinates. The pbe is frequently an integro-differential equation that cannot be solved analytically, and hence a numerical approach must be taken. Analytical approximations can sometimes be derived for steady state solutions at the ends of the distribution (Diemer and Olson, 2002a) but a complete solution can only be found for the simplest (and generally non-physical) cases \textit{(e.g.} Patil and Andrews, 1998; Lage, 2002; McCoy and Madras, 2003). The method of moments \textit{(e.g.} Frenklach, 1985; Frenklach and Harris, 1986, for coagulation only) is an efficient method of solution, but gives no information about the shape of the distribution. In more complicated situations, there can also be problems with closure of the set of equations to be solved (Diemer and Olson, 2002b). Efforts have been made to reconstruct the distribution from the moment solution (Diemer and Olson, 2002b), but there are difficulties, given that there is no obvious orthogonal set of functions from which a distribution can be constructed. Alternatively, an assumed shape can be imposed on the distribution to fit with a limited number of the moments. This can include monodisperse, log-normal (Grosschmidt et al., 2003) or Rosin-Rammler (Angeli and Hewitt, 2000).

Sectional methods are a well established solution technique in population balance modelling. Discretisation of the size distribution can be linear (Hidy and
Brock, 1970) or geometric (Gelbard and Seinfeld, 1978; Hounslow et al., 1988; Litster et al., 1995); a coarser grid is possible with the latter method. To recover functionals of the size distribution, a pivotal particle size must be chosen for each interval.

Recent work includes efforts to use discretisation as a tool for recovering specific functionals of the distribution, rather than merely to approximate the system (Kumar and Ramkrishna, 1996a) and to use a varying pivotal particle size for each discrete interval in order to better account for non-uniformities in number density (Kumar and Ramkrishna, 1996b). Several sectional approaches have been examined and reviewed by Vanni (2000).

An alternative strategy is to employ finite-element methods, in which the solution is approximated as linear combinations of basis functions over a finite number of subdomains. Finite element methods include the method of weighted residuals, the method of orthogonal collocation and Galerkin’s method. Nicmanis and Hounslow (1998) applied a finite-element method to various cases of the steady state population balance equation, finding more accurate solution than using the discretisation method and using less computational power.

In a similar vein, Liu and Cameron (2001) used a wavelet-based method. They found particular success in cases where steep fronts in the distribution are present, whereas discretisation in other solution methods would have to address this problem directly.

A detailed review of numerical techniques for solution of the population balance equation was performed by Ramkrishna (1985), while Kraft (2005) reviewed solution methods for the field of nanoparticle synthesis. Many of the techniques in this latter paper are also applicable to population balances in other areas.

### 1.3 Stochastic Simulation

Broadly speaking, stochastic simulation involves generating fictitious realisations of the behaviour of a set of particles. This particle ensemble is an approximation of the real life system being examined. As the process relies on random number generation to choose the nature and timing of the coalescence and breakage events, many trajectories are generated, and an average (which converges to the solution of the population balance equation (Eibeck and Wagner, 2000a)) is
calculated.

One specific example of such a stochastic process is the so-called Marcus-Lushnikov process (Marcus, 1968; Lushnikov, 1978), in which a population of (initial) size \( n \) evolves by coagulation of pairs of particles at various intervals. The choice of which particles coagulate, and the length of the intervals between coagulation events, depends on the coagulation kernel, \( K(x, y) \). As the number of particles tends to infinity, the Marcus-Lushnikov process converges to the solution of the discrete form of the Smoluchowski coagulation equation (von Smoluchowski, 1916) (see Chapter 2 for explanation of the terms involved):

\[
\frac{\partial}{\partial t} c(x, t) = \frac{1}{2} \sum_{y=1}^{x-1} K(x - y, y) c(x - y, t) c(y, t) - \sum_{y=1}^{\infty} K(x, y) c(x, t) c(y, t). \tag{1.1}
\]

Stochastic (or Monte Carlo) methods are an attractive way of tackling the population balance equation due to the discrete nature of the mechanisms being modelled: breakage and coalescence (Ramkrishna, 1981). They can also provide a feasible solution method for multi-dimensional (dependent on more than just size) population balance problems, where standard numerical techniques can become prohibitively computationally expensive.

For the one-dimensional case (size dependency only), direct stochastic simulation (e.g. Gillespie, 1972, 1975) can be outperformed by conventional numerical techniques. Hence, there are several ways of improving the efficiency of simulation. To avoid the problems of decreasing particle number (and therefore variance increase) or increasing particle number (and thereby reduced efficiency), a particle combining technique can be employed, in which parallel arrays of particles are periodically combined to give an array with the desired number of particles (Sabelfeld et al., 1996). A natural extension of this technique is the constant number simulation method of Matsoukas and co-workers (Smith and Matsoukas, 1998; Lee and Matsoukas, 2000; Lin et al., 2002). One of the major inefficiencies of simulation comes from the coalescence process. A direct consideration of \( \frac{1}{2} n(n-1) \) possible particle pairs (out of \( n \) particles) can be avoided by introducing the fictitious jump technique of Eibeck and Wagner (2000a,b). Here a majorant kernel is used that enables independent generation of the two coalescing particles. The use of a majorant kernel leads to shorter waiting times between events; thus, to ensure that the simulation remains exact, null events (fictitious jumps) must
be included in the simulation. Babovsky (1999) considered a method for variance reduction by considering mass (rather than number) concentration. This mass flow technique has been used by Eibeck and Wagner (2001) to develop a stochastic algorithm that enables accurate simulation with far fewer stochastic particles in the simulation array.

Due to the finite size of a stochastic particle system, it can only be an approximation to a theoretical probability distribution. One important characteristic of a stochastic simulation process is therefore how it converges to the solution of the population balance equation as the number of stochastic particles increases. Bayewitz et al. (1974) considered a related problem: that of a poorly mixed system. In this work, a poorly mixed system was modelled as a number of distinct compartments (each starting with the same number of particles) that are themselves well-mixed, but that do not interact with each other. In effect, considering the behaviour of this system as the size of the compartments increase is equivalent to considering convergence with respect to the number of stochastic particles. Bayewitz et al. (1974) described qualitatively the discrepancies at low particle numbers, but did not quantify their results. More recently, it has been shown in several cases (Kraft and Wagner, 2003; Goodson and Kraft, 2002) that the systematic error of a stochastic solution decreases as the inverse of the number of stochastic particles.

1.4 Multi-dimensional Population Balance Models

Spielman and Levenspiel (1965) used a Monte Carlo method to approach the two-dimensional problem of coalescence and mass transfer in a liquid-liquid system. However, their approach assumed a monodisperse size distribution, with coalescence being immediately followed by breakage to form two identical drops, i.e., coalescence was in fact treated as a mixing event.

Two-dimensional models for coagulation and sintering of nano-particles were proposed by Xiong and Pratsinis (1992). Mühlenweg et al. (2002) examined three approaches of varying complexity to tackle the problem of nano-particle synthesis: a monodisperse model, a one-dimensional (size only) sectional method and a two-dimensional (size and surface area) sectional method. Due to the computational
expense of the sectional method in the two-dimensional case, they concluded that the extra accuracy was not worth the computational effort. Jeong and Choi (2001) reduced the two-dimensional problem to two (size and area respectively) one-dimensional problems, which were solved using sectional methods. Again, the increase in efficiency was considered to compensate for the loss in accuracy. The problem of simultaneous aggregation and sintering has also been approached more successfully using stochastic simulation (Tandon and Rosner, 1999; Morgan et al., 2004).

In the field of wet powder agglomeration, it is accepted that a multi-dimensional approach is necessary to describe the processes involved (Iveson, 2002), but there is a lack of well-developed models. Wauters (2001) proposed a simple three-dimensional model (based on solid content, liquid content and air content), but the basic nature of the model means it is only a first step towards a phenomenological description of the agglomeration process. Some attempts have been made to decompose a two-dimensional (solid and binder content) problem into two one-dimensional problems (Biggs et al., 2003), but further investigation suggests that particle size and binder content are correlated, rendering this approach inaccurate (Reynolds et al., 2004). Madec et al. (2003) discuss the need for multi-dimensional models, and the utility of stochastic methods for their solution. However, they only make preliminary steps towards actually developing multi-dimensional models.

1.5 Motivation

In most cases, population balance models are one-dimensional, i.e. they are size-dependent only. In this case, the well developed deterministic solution methods that exist can often out-perform stochastic solution in terms of efficiency of simulation. When multi-dimensional problems are introduced, the complexity of the problem can render deterministic solution computationally inefficient.

While it is possible to propose models (either empirical or phenomenological) to describe the interactions between particles, the rates and conservation laws of each individual process can not readily be measured. The most convenient experimental data to be measured are often in the form of a distribution of sizes (or other internal co-ordinates), which must be compared with the solution of the
population balance equation. For multi-dimensional problems, the lack of efficient solution method has hindered the development and testing of multi-dimensional models. Empirical descriptions are often used in place of full consideration of the population balance, but these are normally limited to a stipulated range of operation and are not extendable. With the development of a set of tools that allows efficient solution of the multi-dimensional population balance equation, the development of multi-dimensional population balance models will become a feasible task.

In most cases, mathematical literature on the subject of stochastic methods focusses on existence proofs (Norris, 1999; Aldous, 1999) and does not concern itself with the practicality as a solution method. From an engineering point of view, while it is reassuring (and indeed essential) to know that a particular method will provide a solution, more information is needed when choosing a solution method. Comparisons need to be made between different methods: which is faster; which is simpler to implement; which is able to produce the data in the required format? These are all questions that need to be answered, so practical studies of the algorithms involved need to be performed.

1.6 Aims

It is beyond the scope of this work to apply a stochastic solution method to every single application of population balance modelling and compare the results with standard, deterministic methods. Instead, a selection of specific instances where stochastic solution methods are applicable are examined. The overall aim of this work is to demonstrate the applicability (and efficiency) of the stochastic approach to solution of the multi-dimensional population balance equation. Discussion of specific issues that arise should enable the interested reader to construct an equivalent solution algorithm for any problem of interest. Issues that arise and are tackled are as follows:

- Stochastic simulation of discrete processes:
  - Coalescence
  - Breakage
  - Inflow
– Outflow

• Stochastic simulation of continuous processes:
  – Mass Transfer
  – Growth or shrinkage

• Selection of appropriate majorant kernels

• Reconstruction of size distributions from functionals

• Organisation of stochastic particle systems for efficient generation of probability distributions

• Convergence of solution methods

• Ternary (and higher) breakage

Although the aim of this work is to facilitate future model development, most of this work is based on existing models and is not concerned with developing new ones. However, model development in the field of wet powder agglomeration is very limited, so a novel description of the particles and processes involved is attempted.
Chapter 2

The Smoluchowski Coagulation Equation

2.1 Existing Stochastic Algorithms

Much existing work on stochastic simulation of population balances is centered on the Smoluchowski coagulation equation (von Smoluchowski, 1916):

\[
\frac{\partial}{\partial t} c(x,t) = \frac{1}{2} \sum_{y=1}^{x-1} K(x-y,y)c(x-y,t)c(y,t) - \sum_{y=1}^{\infty} K(x,y)c(x,t)c(y,t), \quad (2.1)
\]

with monodisperse initial condition:

\[
c(0, x) = c_0(x) = \begin{cases} 
1 & : x = 1, \\
0 & : \text{otherwise}.
\end{cases} \quad (2.2)
\]

Equation (2.1) describes the time evolution of \(c(x, t)\), the number (per unit volume) of particles of (integer) size \(x\). The first term on the right-hand side describes the rate of production of particles of size \(x\), due to the coagulation of two particles of sizes \(y\) and \((x - y)\). The second term on the right-hand side describes the rate of disappearance of particles of size \(x\), due to coagulation of a particle of size \(x\) with one of any other size. \(K(x, y)\) is the coagulation rate kernel, which describes the rate of coagulation of particles of sizes \(x\) and \(y\). In this (discrete) case, the size of a particle, \(x\), is taken to mean the mass (or volume) of a particle divided by the mass (or volume) of the smallest possible particle. Thus, the size,
$x$, can be considered as the number of monomers contained in a particle, and takes integer values.

Gillespie (1972, 1975) was among the first to propose a stochastic algorithm (see Appendix A, Chapter 4), deriving the necessary forms of waiting time parameters and particle index probability distributions. Although limited by computational power at the time, his work was an effective method of simulating the behaviour of particles in a cloud. Where his work differs from current examination of the Smoluchowski equation is that his formulation was based on an absolute number of particles rather than a number concentration. The Smoluchowski equation can be considered to be the limit as the number of particles tends to infinity and, due to spatial homogeneity, the concentration is an accurate description of the whole population. Bayewitz et al. (1974) considered the implications of varying the number of particles in the particle population. Although this was ostensibly done to consider the effects of mixedness on overall population behaviour, it is equivalent to considering the convergence behaviour of a stochastic algorithm. Due to the limits of computational power at the time, Bayewitz et al. (1974) could only consider a qualitative description of this behaviour.

Gillespie’s algorithm has not become widely used, due to its inefficiency outweighing any benefits of the stochastic approach. Eibeck and Wagner (2000b) improved the efficiency of stochastic simulation by introducing the concept of a majorant kernel and fictitious jumps. In this method, a simplified coagulation kernel is used, such that generation of waiting times and particle index distribution is much more efficient. In order that the simulation remains exact, null events are included in the simulation, such that the overall number of coagulation events remains the same.

2.2 The Stochastic Model

2.2.1 Markov Process With Fictitious Jumps

To simulate the coagulation process in Equation (2.1), we follow the method introduced by Eibeck and Wagner (2000b), using a Markov Process with fictitious jumps.

For a detailed description of the method, we refer to section 2 of Eibeck and Wagner (2000b), and give a brief summary of the motivation and method here.
Consider a stochastic particle system:

\[ x_i(t), \quad i = 1, 2, \ldots, n, \quad t \geq 0, \tag{2.3} \]

where \( x_i(t) \) represents the size of particle \( i \) at time \( t \), and \( n \) is the (time varying in this case) number of particles in the system. In this (discrete) case, the particle sizes only take integer values. The initial system is chosen so as to approximate the initial condition (2.2).

An approximation to a measure valued version of (2.1) is sought; our measure valued solution is denoted \( P^N(x,t) \) and is a sequence of jump processes, \( i.e. \), a series of random variables, whose subsequent state depends only on the current state, not on its history. It is a Markov process. The realisation of any of this sequence of random variables is denoted \( p(x) \), which indicates a general representation of the state of the system at any point. This realisation takes the form:

\[ p(x) = \frac{1}{N} \sum_{i=1}^{n} \delta(x - x_i), \quad x_i > 0, \quad n = 1, 2, \ldots \tag{2.4} \]

Using this representation, the concentration, \( c(x,t) \), is approximated by:

\[ c(x,t) \sim 1 \frac{1}{N} \# \{ i : x_i(t) = x \}. \tag{2.5} \]

From this relation, it can be seen that the parameter \( N \), known as the particle number, can be considered to be the equivalent of a normalisation parameter or sample size. Any other functionals of interest (\( e.g. \) moments) can be approximated by:

\[ F(t) = \int_{0}^{\infty} \phi(x)c(x,t)dx \sim \int_{0}^{\infty} \phi(x) \frac{1}{N} \sum_{i=1}^{n} \delta(x - x_i)dx \sim \frac{1}{N} \sum_{i=1}^{n} \phi(x_i) \tag{2.6} \]

The subsequent coagulation of particles is a Markov Process in which individual coagulation events are separated by an exponentially distributed waiting time (Gillespie, 1975). The waiting time, \( \tau \), is generated according to:

\[ \text{Prob}\{\tau(p) \geq s\} = \exp(-\rho K(p) \cdot s), \quad s \geq 0, \tag{2.7} \]
where $\rho_K(p)$ is the waiting time parameter (for the coagulation process), given by

$$\rho_K(p) = \frac{1}{2N} \sum_{1 \leq i \neq j \leq n} K(x_i, x_j). \quad (2.8)$$

For each event, particle indices $i$ and $j$ must be chosen according to the coagulation rate kernel, i.e., according to the probabilities:

$$\frac{K(x_i, x_j)}{\sum_{1 \leq i \neq j \leq n} K(x_i, x_j)}, \quad (2.9)$$

and then the system makes the jump:

$$p(x) \rightarrow p(x) - \frac{\delta(x - x_i)}{N} - \frac{\delta(x - x_j)}{N} + \frac{\delta(x - x_i - x_j)}{N}, \quad (2.10)$$

i.e., particles of sizes $x_i$ and $x_j$ are removed and a particle of size $x_i + x_j$ is added.

Generation of the joint probability distribution of $i$ and $j$ requires summing over a number of terms of the order of $n^2$, so a majorant kernel, $\hat{K}(x_i, x_j)$, is introduced, according to:

$$K(x_i, x_j) \leq \hat{K}(x_i, x_j), \quad x_i, x_j > 0, \quad (2.11)$$

such that the waiting time, $\tau$, is then efficiently generated with the parameter:

$$\hat{\rho}_K(p) = \frac{1}{2N} \sum_{1 \leq i \neq j \leq n} \hat{K}(x_i, x_j), \quad (2.12)$$

and the joint probability distribution:

$$\frac{\hat{K}(x_i, x_j)}{\sum_{1 \leq i \neq j \leq n} \hat{K}(x_i, x_j)} \quad (2.13)$$

enables independent generation of the indices $i$ and $j$, reducing the computational expense to the order of $n$.

In order that this method still gives convergence to the solution of a measure valued version of (2.1) in the limit $N \rightarrow \infty$, fictitious jumps are introduced.
These are additional, null events that occur with probability

$$1 - \frac{\hat{K}(x_i, x_j)}{\hat{K}(x_i, x_j)}$$ (2.14)

when the indices $i$ and $j$ have been chosen.

### 2.2.2 Organisation Of Particle System

For efficient simulation and generation of size dependent distributions, the stochastic particle sizes $(x_1, x_2 \ldots)$ are stored in a series of arrays (or bins) as described by Eibeck and Wagner (2000b). The particles are organised into $\psi$ groups, i.e., their sizes are denoted by

$$y_{z,k}, \quad z = 1, \ldots, \psi, \quad k = 1, \ldots, \alpha_z.$$ (2.15)

so that

$$b_{z-1} < y_{z,k} \leq b_z, \quad \forall z = 1, \ldots, \psi \quad k = 1, \ldots, \alpha_z,$$ (2.16)

where

$$0 \equiv b_0 < b_1 < \ldots < b_\psi$$ (2.17)

and

$$x_{\text{max}} \leq b_\psi.$$ (2.18)

$x_{\text{max}}$ is the upper bound for the particle size, which will depend on the initial condition and the inflow rate.

The group size bounds, $b_z$, are chosen to be related by

$$b_z = \beta^{z-1}, \quad z = 1, 2, \ldots,$$ (2.19)

where $\beta > 1$.

Then, in order to generate the distribution

$$\frac{x_j^x}{\sum_{i=1}^{n} x_i^x}, \quad j = 1, \ldots, n,$$ (2.20)

the choice of group can be made by the discrete inversion (linear search) method, while the choice of particle index within the group can be done by acceptance-
rejection, with a minimum efficiency of $1/\beta^\varepsilon$.

### 2.2.3 Majorant Kernel

The discrete form of the Smoluchowski equation is particularly useful in applications where the particle sizes take discrete values, *i.e.*, they are measured in numbers of monomers. One such application is the field of nano-particle synthesis in flames (see Chapter 5).

A typical coagulation rate kernel, in this case for coagulation of nano-particles in the free molecular regime, is given by:

$$K(x, y) = \left(\frac{3}{4\pi}\right)^{\frac{1}{2}} \left(\frac{8\pi k_B T}{\rho}\right)^{\frac{1}{2}} \left(\frac{m_1}{\rho}\right)^{\frac{1}{6}} \left(\frac{1}{x} + \frac{1}{y}\right)^{\frac{1}{2}} \left(x^{\frac{1}{3}D_F} + y^{\frac{1}{3}D_F}\right)^2.$$  \hfill (2.21)

$D_F$ is a parameter describing the shape of the particles involved.

In the subsequent analysis, we use the dimensionless form of this kernel:

$$K(x, y) = \left(\frac{1}{x} + \frac{1}{y}\right)^{\frac{1}{2}} \left(x^{\frac{1}{3}D_F} + y^{\frac{1}{3}D_F}\right)^2.$$  \hfill (2.22)

Note that the function

$$\tilde{c}(x, t) = \alpha_2 c(x, \alpha_1 \alpha_2 t), \quad t \geq 0, \quad x > 0, \quad \alpha_1, \alpha_2 \geq 0,$$  \hfill (2.23)

solves Equation (2.1) with the kernel $\tilde{K} = \alpha_1 K$ instead of $K$ and initial condition $\tilde{c}_0 = \alpha_2 c_0$ instead of $c_0$. The scaling factors

$$\alpha_1 = \left(\frac{3}{4\pi}\right)^{\frac{1}{2}} \left(\frac{8\pi k_B T}{\rho}\right)^{\frac{1}{2}} \left(\frac{m_1}{\rho}\right)^{\frac{1}{6}}$$  \hfill (2.24)

and $\alpha_2$, which will be specific to a given situation, can be calculated after a general simulation takes place, and the dimensionless time and concentration can be scaled by these factors to give a dimensional answer.

A linear majorant kernel has been proposed by Eibeck and Wagner (2001) for this kernel for the case $D_F = 3$ (spherical particles), namely:

$$\sqrt{\frac{1}{x} + \frac{1}{y}} \left(x^{1/3} + y^{1/3}\right)^2 \leq c(x + y), \quad c > 0.$$  \hfill (2.25)
However, this majorant has several drawbacks, especially when we try to apply it to the case of $D_F < 3$ (non-spherical particles), stemming from the fact that it does not have the same degree of homogeneity as the coagulation kernel itself.

Recall that a kernel is homogeneous with degree $\gamma$ if:

\[
K(\lambda x, \lambda y) = \lambda^\gamma K(x, y),
\]  

(2.26)

So in this case, the majorant is homogeneous with degree 1, and the kernel (2.21) itself is homogeneous with degree $\frac{2}{D_F} - \frac{1}{2}$.

It can easily be shown that the linear majorant (2.25) is only valid for $D_F \geq 2$ by considering the leading order terms of the kernel and its linear majorant. We have, for any $x, y$:

\[
\sqrt{\frac{1}{x} + \frac{1}{y}} (x^{1/D_F} + y^{1/D_F})^2 \geq \sqrt{\frac{1}{y} x^{2/D_F}}.
\]  

(2.27)

Clearly for $D_F < 2$ this RHS increases faster than linearly with increasing $x$ and there is not a value of $c$ such that the required majorant inequality (2.11) holds for all $x$. Typical values for $D_F$ for non-spherical particles are between 1.7 and 2.5 (Matsoukas and Friedlander, 1991), so for the general case a new majorant kernel must be found.

Even for the case $D_F \geq 2$ we encounter problems. The value of the constant $c$ depends on the minimum value, $x_{min}$, that the size can take:

\[
c = 2\sqrt{2} x_{min}^{\frac{2}{D_F} - \frac{3}{2}}.
\]  

(2.28)

It is then clear that the acceptance efficiency of the majorant:

\[
\frac{K(x, y)}{\hat{K}(x, y)}
\]  

(2.29)

is also dependent on the minimum size. For $y = x$ we find:

\[
\frac{K(x, x)}{\hat{K}(x, x)} = \left( \frac{x}{x_{min}} \right)^{\frac{2}{D_F} - \frac{3}{2}},
\]  

(2.30)
and for \( y = x_{\text{min}} \), we find, in the limit \( x \gg y \):

\[
\lim_{x \to \infty} \frac{K(x, x_{\text{min}})}{K(x, x_{\text{min}})} = \lim_{x \to \infty} \frac{1}{2\sqrt{2}} \left( \frac{x}{x_{\text{min}}} \right)^{\frac{2}{1/2}}^{-1}.
\]  

(2.31)

These both have a lower limit of zero.

In our case, \( x_{\text{min}} = 1 \), so there is a well defined minimum size, which is equal to the size of the particles in the initial size distribution \( c_0(x) \) and is therefore of the same order of magnitude as most of the particles in the system, at least for low simulation times. However, there are situations in which this is not the case; the ratio \( \frac{x}{x_{\text{min}}} \) is then correspondingly larger, and much lower acceptance efficiency is encountered. For example, in the study of oxidation kinetics in flames, it may be desirable to seed a flame with particles much larger than the minimum size available.

In cases like this, the minimum particle size has a direct bearing on the efficiency of the majorant kernel, and therefore on the efficiency of simulation. We seek a majorant kernel that is homogeneous with the same degree as the actual coagulation kernel.

A majorant kernel is constructed by using the inequalities:

\[
(a + b)^{\varepsilon} \leq 2^{\varepsilon-1} (a^{\varepsilon} + b^{\varepsilon}), \quad \varepsilon \geq 1, \quad a, b > 0,
\]  

(2.32) and

\[
(a + b)^{\varepsilon} \leq (a^{\varepsilon} + b^{\varepsilon}), \quad 0 < \varepsilon < 1, \quad a, b > 0.
\]  

(2.33)

For a proof of these, see Appendix A.

We now have:

\[
\left( \frac{1}{x} + \frac{1}{y} \right)^{\frac{1}{2}} \left( x^{\frac{1}{2}P} + y^{\frac{1}{2}P} \right)^{2} \leq 2 \left( x^{-\frac{1}{2}} + y^{-\frac{1}{2}} \right) \left( x^{\frac{1}{2}P} + y^{\frac{1}{2}P} \right)
\]  

(2.34)

In fact, the maximum value of the ratio \( \frac{K}{K} \) is less than one, so this majorant is not as efficient as it could be. By examining the maximum value of \( \frac{K}{K} \), we find that it is more useful to use (see Appendix B for justification):

\[
\left( \frac{1}{x} + \frac{1}{y} \right)^{\frac{1}{2}} \left( x^{\frac{1}{2}P} + y^{\frac{1}{2}P} \right)^{2} \leq \sqrt{2} \left( x^{-\frac{1}{2}} + y^{-\frac{1}{2}} \right) \left( x^{\frac{1}{2}P} + y^{\frac{1}{2}P} \right)
\]  

(2.35)
in the case $1.7 < D_F < 2.5$, and

$$
\left( \frac{1}{x} + \frac{1}{y} \right)^{\frac{2}{3}} \left( x^{\frac{1}{2}} + y^{\frac{1}{2}} \right)^2 \leq 1.4178 \left( x^{-\frac{1}{2}} + y^{-\frac{1}{2}} \right) \left( x^{\frac{1}{2}} + y^{\frac{1}{2}} \right)
$$

(2.36)

for $D_F = 3$. Both of these are clearly more efficient majorant kernels.

Now when we consider the acceptance efficiency of the majorant, we find, for $x = y$:

$$
\frac{K(x,x)}{\hat{K}(x,x)} = \begin{cases} 
1 & : 1.7 < D_F < 2.5, \\
\sqrt{2} & : D_F = 3,
\end{cases}
$$

(2.37)

and for $y = x_{\min}$, $x \gg y$ we find:

$$
\lim_{x \to \infty} \frac{K(x,x_{\min})}{\hat{K}(x,x_{\min})} = \begin{cases} 
\frac{1}{\sqrt{2}} & : 1.7 < D_F < 2.5, \\
\frac{1}{1.4178} & : D_F = 3.
\end{cases}
$$

(2.38)

Because the kernel (2.21) and its majorant (2.35) are homogeneous with the same degree, the dependence of the acceptance efficiency on the minimum size is removed. Also, we find that we can use this new majorant for the case $D_F < 2$.

The majorant kernel can be multiplied out to give

$$
\hat{K}(x_i,x_j) = \sqrt{2} \left( x_i^{\frac{2}{D_F}} x_j^{\frac{2}{D_F}} \right)^{\frac{1}{2}} + x_i^{\frac{2}{D_F}} x_j^{\frac{2}{D_F}} + x_i^{\frac{2}{D_F}} x_j^{\frac{2}{D_F}} + x_i^{\frac{2}{D_F}} x_j^{\frac{2}{D_F}}
$$

(2.39)

which can be considered as being composed of

$$
\hat{K} = \hat{K}_1 + \hat{K}_2 + \hat{K}_3 + \hat{K}_4.
$$

(2.40)

The choice of which of these majorant components to use to generate indices $i$ and $j$ is then made probabilistically, as described in section 2.2 of Eibeck and Wagner (2000b).

It should be noted that

$$
\hat{K}_1(x_i,x_j) = \sqrt{2}(x_i^{\frac{2}{D_F}}) \quad \text{and} \quad \hat{K}_2(x_i,x_j) = \sqrt{2}(x_j^{\frac{2}{D_F}})
$$

give identical distributions for $i$ and $j$, except with the indices swapped. The same is true for $\hat{K}_3$ and $\hat{K}_4$. Because the coagulation step of the stochastic algorithm is symmetrical with respect to the indices $i$ and $j$, the order in which
they are generated does not matter. Therefore, for the purposes of generating the particle indices, we can consider just two different functions, $\hat{K}_1$ and $\hat{K}_3$. Thus we calculate a value of $\hat{\rho}(p)$ associated with each kernel component according to:

$$
\hat{\rho}_1(p) = \frac{1}{2N} \sum_{1 \leq i \neq j \leq n} \sqrt{2(x_i^2F - \frac{1}{2})} \quad \text{and} \quad \hat{\rho}_3(p) = \frac{1}{2N} \sum_{1 \leq i \neq j \leq n} \sqrt{2(x_i^2F x_j^{-\frac{1}{2}})},
$$

(2.41)

and choose $\hat{K}_1$ (or otherwise $\hat{K}_3$) to generate indices $i$ and $j$ with probability:

$$
\frac{\hat{\rho}_1(p)}{\hat{\rho}_1(p) + \hat{\rho}_3(p)}.
$$

(2.42)

Note that we could have simply multiplied out the second bracket of (2.22) rather than using the inequality (2.32) and obtained:

$$
\hat{K} = x^{\frac{1}{2F} - \frac{1}{2}} + y^{\frac{1}{2F} - \frac{1}{2}} + x^{\frac{1}{2F} y^{-\frac{1}{2}}} + x^{-\frac{1}{2}} y^{-\frac{1}{2}} + 2x^{\frac{1}{2F} - \frac{1}{2}} y^{\frac{1}{2F}} + 2x^{\frac{1}{2F}} y^{\frac{1}{2F} - \frac{1}{2}}
$$

(2.43)

However, using this kernel would require at each coagulation step the storage and update of two more quantities:

$$
\sum_{i=1}^{n} x_i^{\frac{1}{2F} - \frac{1}{2}} \quad \text{and} \quad \sum_{i=1}^{n} x_i^{\frac{1}{2F}}
$$

which would reduce the efficiency of the simulation.

This analysis has been performed for the free-molecular regime kernel. Inspection of the continuum regime kernel (see Chapter 5) reveals that no majorant is necessary, as it can be simply multiplied out to give a number of kernels.

**Appendices**

**A Justification Of Choice Of Majorant Kernel**

To prove that:

$$
(x + y)^n \leq 2^{n-1} (x^n + y^n) \quad n > 1, \quad x, y > 0,
$$

(2.44)
consider the function:

\[
z(x, y) = 2^{n-1} (x^n + y^n) - (x + y)^n \\
= x^n \left[ 2^{n-1} \left( 1 + \left( \frac{y}{x} \right)^n \right) - \left( 1 + \frac{y}{x} \right)^n \right] \\
= x^n \left[ 2^{n-1} \left( 1 + f_y^n \right) - (1 + f_y)^n \right], \tag{2.45}
\]

where \( f_y = \frac{y}{x} \).

We require:

\[
z(f_y) = 2^{n-1} \left( 1 + f_y^n \right) - (1 + f_y)^n \geq 0 \quad \forall f_y > 0. \tag{2.46}
\]

Now,

\[
\frac{dz}{df_y} = n(2f_y)^{n-1} - n(1 + f_y)^{n-1} = 0 \quad \text{at } f_y = 1. \tag{2.47}
\]

\[
\frac{d^2z}{df_y^2} = 2n(n-1)(2f_y)^{n-2} - n(n-1)(1 + f_y)^{n-2} \\
= n(n-1)2^{n-2} > 0 \quad \text{at } f_y = 1. \tag{2.48}
\]

Giving a global minimum at \( f_y = 1 \), or \( x = y \).

At \( f_y = 1 \), the minimum value of \( z \) is:

\[
z(1) = (2^{n-1} \times 2) - 2^n = 0. \tag{2.49}
\]

Now, to prove that:

\[(x + y)^n \leq (x^n + y^n) \quad 0 < n < 1, \quad x, y > 0 \tag{2.50}\]

We require:

\[(1 + f_y)^n \leq 1 + f_y^n, \tag{2.51}\]

where \( f_y = \frac{y}{x} \) as before.

We have, for \( 0 < n < 1 \) and \( 0 < f_y < 1 \):

\[
(1 + f_y)^n \leq 1 + f_y, \tag{2.52}
\]

\[
1 + f_y^n \geq 1 + f_y. \tag{2.53}
\]
Therefore:

\[(1 + f_y)^n \leq 1 + f^n_y, \quad (2.54)\]

as required.

In the case \(f_y > 1\) we can rearrange:

\[(1 + f_y)^n \leq 1 + f^n_y \quad (2.55)\]

to give

\[\left(1 + \frac{1}{f_y}\right)^n \leq 1 + \left(\frac{1}{f_y}\right)^n, \quad (2.56)\]

and the previous argument still holds.

In the case of \(n = 1\), we have (trivially)

\[(x + y)^n = 2^{n-1}(x^n + y^n) = x^n + y^n = x + y, \quad (2.57)\]

so no majorant kernel is required.

\section{Justification Of Majorant Kernel Multiplication Factor}

The first majorant kernel considered in Section 2.2.3 is

\[
\left(\frac{1}{x} + \frac{1}{y}\right)^\frac{1}{2} \left(x^{\frac{1}{DF}} + y^{\frac{1}{DF}}\right)^2 \leq 2 \left(x^{-\frac{1}{2}} + y^{-\frac{1}{2}}\right) \left(x^{\frac{2}{DF}} + y^{\frac{2}{DF}}\right). \quad (2.58)
\]

For an efficient majorant kernel we expect:

\[
\max_{x,y} \frac{K(x, y)}{\hat{K}(x, y)} = \max_{x,y} \frac{\left(\frac{1}{x} + \frac{1}{y}\right)^\frac{1}{2} \left(x^{\frac{1}{DF}} + y^{\frac{1}{DF}}\right)^2}{2 \left(x^{\frac{2}{DF}} + y^{\frac{2}{DF}}\right)^\frac{1}{2} \left(x^{-\frac{1}{2}} + y^{-\frac{1}{2}}\right) + x^{-\frac{1}{2}}y^{\frac{2}{DF}} + x^{\frac{2}{DF}}y^{-\frac{1}{2}}} = 1. \quad (2.59)
\]
In fact, we can rewrite the above expression as:

\[
\max_{x,y} \frac{K(x, y)}{K(x, y)} = \max_{f_y} \frac{z(f_y)}{f_y} = \max_{f_y} \frac{\left(1 + \frac{1}{f_y}\right)^{\frac{1}{2}} \left(1 + f_y^{\frac{1}{2}}\right)^2}{2 \left(1 + f_y^{\frac{1}{2}} - \frac{1}{2} + f_y^{-\frac{1}{2}} + f_y^{\frac{1}{2}}\right)}, \quad 0 < f_y < 1.
\]

(2.60)

We wish to find the maximum value of this function, and if it is less than 1, alter the multiplication factor to achieve maximum efficiency.

Splitting up \(z(f_y)\) into two functions gives

\[
z(f_y) = z_1(f_y) \times z_2(f_y) = \left(1 + \frac{1}{f_y}\right)^{\frac{1}{2}} \times \frac{\left(1 + f_y^{\frac{1}{2}}\right)^2}{2 \left(1 + f_y^{\frac{1}{2}}\right)}.
\]

(2.61)

These two functions are monotonically varying functions between 0 and 1, with \(z_1(f_y)\) decreasing from 1 to \(\frac{1}{\sqrt{2}}\) and \(z_2(f_y)\) increasing from \(\frac{1}{2}\) to 1.

Due to the symmetric nature of \(z(f_y)\), there must be a turning point at \(f_y = 1\), where \(z(f_y) = \frac{1}{\sqrt{2}}\), thus, depending on the value of \(D_F\) we have two cases of interest:

- \(z(f_y)\) is monotonically increasing as \(f_y\) varies between 0 and 1, so the maximum occurs at \(f_y = 1\). This is characterised by the second derivative of \(z(f_y)\) at \(r = 1\) being negative.

- \(z(f_y)\) contains a local maximum on the range \(0 < f_y < 1\) which may be greater than \(z(1)\). This is characterised by \(z(1)\) being a local minimum and the second derivative at this point being positive.

So, by differentiating \(z(f_y)\) twice with respect to \(f_y\) and setting \(f_y = 1\), we obtain a function in \(D_F\) that can be solved numerically to distinguish between the two cases above. The crossover occurs at \(D_F \approx 2.82843\), with the second derivative being negative for \(D_F < 2.82843\) and positive for \(D_F > 2.82843\). For non-spherical particles, typical values for \(D_F\) are between 1.7 and 2.5 (Matsoukas and Friedlander, 1991), so the first case applies and

\[
\max_{f_y} z(f_y) = \frac{1}{\sqrt{2}}.
\]

(2.62)
So to make the majorant kernel more efficient, it can be multiplied by \( \frac{1}{\sqrt{2}} \), to give the multiplication factor as \( \sqrt{2} \) instead of 2.

If spherical particles are considered, \( D_F = 3 \) and the second case applies. Differentiating \( z(f_y) \) and setting it to zero to find the maximum value gives \( \max_{f_y} z(f_y) \approx 0.708895 \), so the factor of two becomes a factor of \( 2 \times 0.708895 = 1.41778 \).
Chapter 3

Sources and Sinks

It is rare that the Smoluchowski coagulation equation is sufficient to describe all the processes in a given physical situation. Other processes, such as inflow, outflow and breakage are common, and on extension to the multi-dimensional case, other processes such as mass transfer may also need to be considered. A source term can be required in the case of flame aerosol synthesis to model chemical reaction producing monomer particles and inflow/outflow can be important processes in liquid-liquid extraction systems.

The population balance equation when only inflow and outflow are considered looks like:

$$\frac{\partial}{\partial t}c(x,t) = \frac{c_{in}(x,t) - c(x,t)}{\tau_{res}},$$

(3.1)

with initial condition

$$c(x,0) = c_0(x).$$

(3.2)

To be able to stochastically simulate inflow and outflow, we must derive the appropriate forms for the waiting time parameter and jump term.

3.1 Outflow

Consider the gaseous case, i.e., with constant volume. For the special case of $c_{in}(x) = 0$, i.e., no inflow, we can define the adjoint operator:

$$\mathcal{A}^*c(x,t) = -\frac{1}{\tau_{res}}c(x,t),$$

(3.3)
from which we can obtain $A\Phi(x)$:

$$\int A^*\phi(x)\Phi(x)\,dx = -\frac{1}{\tau_{res}} \int \phi(x)\Phi(x)\,dx = \int \phi(x)(A\Phi)(x)\,dx. \quad (3.4)$$

Therefore we have

$$A\Phi(x) = -\frac{1}{\tau_{res}} \Phi(x). \quad (3.5)$$

This suggests the infinitesimal generator is self-adjoint, and we can rewrite it as:

$$A_{out}^N\Phi(p) = -\frac{1}{\tau_{res}} \Phi(p)$$

$$= -\frac{1}{\tau_{res}} \int \phi(x)p(x)\,dx$$

$$= -\frac{1}{\tau_{res}} \int \phi(x) \frac{1}{N} \sum_{i=1}^n \delta(x-x_i)\,dx$$

$$= -\frac{1}{\tau_{res}} \sum_{i=1}^n \frac{1}{N} \int \phi(x)\delta(x-x_i)\,dx$$

$$= \frac{1}{\tau_{res}} \sum_{i=1}^n \left[ \int \phi(x)p(x)\,dx - \frac{1}{N} \int \phi(x)\delta(x-x_i)\,dx - \int \phi(x)p(x)\,dx \right]$$

$$= \sum_{i=1}^n \frac{1}{\tau_{res}} [\Phi(J_{out}(p,i)) - \Phi(p)]$$

$$= \sum_{i=1}^n K(x_i) [\Phi(J_{out}(p,i)) - \Phi(p)], \quad (3.6)$$

with

$$J_{out}(p,i) = p - \frac{\delta(x-x_i)}{N}, \quad (3.7)$$

and

$$K_{out}(x_i) = \frac{1}{\tau_{res}}. \quad (3.8)$$

This suggests the following simulation algorithm

1. Generate the initial state $P^N(x,0) = p(x)$. 

\[CHAP\]
2. Wait an exponentially distributed time step with parameter

\[ \rho_{\text{out}}(p) = \sum_{i=1}^{n} K(x_i) = \frac{n}{\tau_{\text{res}}}. \]  

(3.9)

3. Generate a discrete uniform random deviate, \( i \), on the set \( 1, 2, \ldots, n \).

4. Remove particle at position \( i \) from the system.

5. Go to step 2.

3.2 Inflow

For inflow, we have the adjoint operator:

\[ A^* c(x, t) = \frac{1}{\tau_{\text{res}}} c_{\text{in}}(x, t). \]  

(3.10)

3.2.1 Stochastic Source

Inflow can be treated stochastically in two ways. Consider modelling a multi-cell liquid-liquid contacting device (see Chapter 7), in which each cell is modelled by a stochastic particle system. Flow of droplets into a cell from an adjacent one would be modelled as a source term, with the inflowing particle size distribution being the same as the size distribution in the adjacent cell.

We model the inflowing density as a stochastic particle system, \( x_{\text{in},i}(t), \quad i = 1, \ldots, n_{\text{in}} \). Equation (3.10) suggests that the infinitesimal generator for inflow is, like that for outflow, self-adjoint. We can rewrite it as:
\[ A_{in}^{N} \Phi(p) = \frac{1}{\tau_{res}} \Phi(p_{in}) \]
\[ = \frac{1}{\tau_{res}} \int \phi(x)p_{in}(x)dx \]
\[ = \frac{1}{\tau_{res}} \int \phi(x) \frac{1}{N} \sum_{i=1}^{n_{in}} \delta(x - x_{in,i})dx \]
\[ = \frac{1}{\tau_{res}} \sum_{i=1}^{n_{in}} \frac{1}{N} \int \phi(x)\delta(x - x_{in,i})dx \]
\[ = \sum_{i=1}^{n_{in}} \frac{1}{\tau_{res}} \left[ \int \phi(x)p(x)dx + \frac{1}{N} \int \phi(x)\delta(x - x_{in,i})dx - \int \phi(x)p(x)dx \right] \]
\[ = \sum_{i=1}^{n_{in}} \frac{1}{\tau_{res}} \left[ \Phi(J_{in}(p,i)) - \Phi(p) \right] \quad (3.11) \]

The simulation algorithm is now:

1. Generate the initial state \( P^{N}(x,0) = p(x) \).

2. Wait an exponentially distributed time step with parameter

\[ \rho_{out}(p) = \sum_{i=1}^{n_{in}} K(x_{in,i}) = \sum_{i=1}^{n_{in}} \frac{1}{\tau_{res}} = \frac{n_{in}}{\tau_{res}}. \quad (3.12) \]

3. Generate a discrete uniform random deviate, \( i \), on the set \( 1, 2, \ldots, n_{in} \).

4. Add a particle of size \( x_{in,i} \) to the system.

5. Go to step 2.

### 3.2.2 Continuous Source Distribution

An alternative stochastic treatment can be used when the source of particles entering the system has a known (or assumed) size distribution; this could be e.g. monodisperse, in the case of flame aerosol synthesis (Ulrich, 1971), normal, in the case of liquid-liquid extraction (Attarakih et al., 2006), or a measured distribution with an assumed shape fit (Simon, 2004). We can then use this size
distribution to directly generate appropriate size particles to add to the stochastic particle system.

Again, using the self-adjoint infinitesimal operator, we can write:

\[
\mathcal{A}^{iN} \Phi(p) = \frac{1}{\tau_{res}} \Phi(c_{in}) \\
= \int \frac{1}{\tau_{res}} c_{in}(y,t) \phi(y) dy \\
= \int \frac{1}{\tau_{res}} c_{in}(y,t) \int \phi(x) \delta(x-y) dx \ dy \\
= \int \frac{N}{\tau_{res}} c_{in}(y,t) \times \\
\left[ \int \phi(x)p(x) dx + \frac{1}{N} \int \phi(x)\delta(x-y) dx - \int \phi(x)p(x) dx \right] dy \\
= \int \frac{N}{\tau_{res}} \left[ \Phi(J_{in}(p,y)) - \Phi(p) \right] c_{in}(y,t) dy. \tag{3.13}
\]

The simulation algorithm is then:

1. Generate the initial state \( P^{N}(x,0) = p(x) \).
2. Wait an exponentially distributed time step with parameter

\[
\rho_{out}(p) = \int K(y)c_{in}(y,t)dy = \int \frac{N}{\tau_{res}} c_{in}(y,t)dy = \frac{n_{in}}{\tau_{res}}. \tag{3.14}
\]
3. Generate a random deviate, \( y \) according to the distribution

\[
\frac{c_{in}(y,t)}{\int_{0}^{\infty} c_{in}(y,t)dy} = \frac{Nc_{in}}{n_{in}}.
\]
4. Add a particle of size \( y \) to the system.
5. Go to step 2.

Note that in Equation (3.14), in order to demonstrate the equivalence of the two algorithms, we have related the parameter \( n_{in} \) to the zeroth moment of the
inflowing size distribution. For any stochastic particle system, we can write:

\[ m_0(t) = \int_0^\infty c(x,t)dx \simeq \int_0^\infty p(x)dx \simeq \int_0^\infty \frac{1}{N} \sum_i^n \delta(x-x_i)dx \]
\[ \simeq \frac{1}{N} \sum_i^n \int_0^\infty \delta(x-x_i)dx \simeq \frac{n}{N}. \quad (3.15) \]

Thus we can, by analogy, use:

\[ m_{in,0}(t) = \int_0^\infty c_{in}(x,t)dx \simeq \frac{n_{in}}{N}. \quad (3.16) \]

### 3.3 Test Case Simulation

In order to test our derived stochastic solution algorithms, we need to compare the results of stochastic simulation to the solution of a known test case. The solution of equation (3.1) is:

\[ c(x,t) = \left(1 - e^{-\frac{t}{\tau_{res}}}\right) c_{in}(x) + e^{-\frac{t}{\tau_{res}}} c_0(x) \quad (3.17) \]

In the case of no outflow, the solution is:

\[ c(x,t) = c_0(x) + \frac{t}{\tau_{res}} c_{in}(x) \quad (3.18) \]

This solution has been stochastically simulated for four cases:

1. There is no outflow, so only inflow is considered. The initial condition is a monodisperse distribution of size 1 and the inflowing density is monodisperse of size 2.

2. There is no inflow, so only outflow is considered. The initial condition is a monodisperse distribution of size 1.

3. Both inflow and outflow are considered, with the same distributions as in case 1.

In these three cases the residence time, \( \tau_{res} \), is 2 and \( N = n_{in} \).
4. We also consider a further case; inflow and outflow with:

\[ c_{in}(x) = \begin{cases} 
2 & x = 2, \\
0 & \text{otherwise.}
\end{cases} \]  \hspace{1cm} (3.19)

\[ c_{0}(x) = \begin{cases} 
1 & x = 1, \\
0 & \text{otherwise.}
\end{cases} \]  \hspace{1cm} (3.20)

In this fourth case, \( \tau_{res} = 2 \) but we now have \( N \neq n_{in} \).

The results of stochastic simulation are shown in Figures 3.1–3.4. The distributions have been simulated and the zeroth, first and second moments have been calculated. In all cases, very good agreement with the analytical solution is found, for a confidence interval of 99.9% and particle number \( N = 100000 \).

### 3.4 Presentation of Results

Stochastic simulation involves a fictitious realisation of the behaviour of a set of particles. Repetition of this process results in an independent set of realisations, which can be treated statistically to provide information about the mean behaviour and the variance. Following Kraft and Wagner (2003), confidence intervals are calculated as follows:

Typical macroscopic properties, such as moments of the particle size distribution, are functionals of the form:

\[ F(t) = \int_{0}^{\infty} \phi(x)c(x,t)dx. \]  \hspace{1cm} (3.21)

These functionals are approximated (as \( N \to \infty \)) by the random variable \( \Xi^{(N)}(t) \), whose realisations (from \( L \) independent repetitions) are given by:

\[ \xi^{(N,l)}(t) = \frac{1}{N} \sum_{i=1}^{n(t)} \phi(x_{i}(t)), \quad l = 1, 2, \ldots, L. \]  \hspace{1cm} (3.22)

For finite \( N \), these realisations exhibit fluctuations, due to the stochastic nature of the processes involved. By performing repeated runs, we can calculate the
Figure 3.1: Moments of the particle size distribution for case 1 (inflow only).
Figure 3.2: Moments of the particle size distribution for case 2 (outflow only).
Figure 3.3: Moments of the particle size distribution for case 3 (inflow and outflow).
Figure 3.4: Moments of the particle size distribution for case 4 (inflow and outflow with $N \neq n_{in}$).
empirical mean value of $\Xi^{(N)}(t)$:

$$\eta_1^{(N,L)}(t) = \frac{1}{L} \sum_{l=1}^{L} \xi^{(N,l)}(t),$$

(3.23)

which is used to approximate the functional of interest (3.21).

In order to construct confidence bounds for $\eta_1^{(N,L)}(t)$, we need to estimate the variance of $\Xi^{(N)}(t)$, given by:

$$\text{Var} \, \Xi^{(N)}(t) \equiv \mathbb{E} \left[ \Xi^{(N)}(t) - \mathbb{E} \Xi^{(N)}(t) \right]^2 = \mathbb{E} \left[ \Xi^{(N)}(t) \right]^2 - \left[ \mathbb{E} \Xi^{(N)}(t) \right]^2.$$  

(3.24)

This quantity is estimated by the empirical variance, defined as

$$\eta_2^{(N,L)}(t) = \frac{1}{L} \sum_{l=1}^{L} \left[ \xi^{(N,l)}(t) \right]^2 - \left[ \eta_1^{(N,L)}(t) \right]^2.$$  

(3.25)

The random variable

$$\frac{\eta_1^{(N,L)}(t) - E\Xi^{(N)}(t)}{\sqrt{\text{Var} \, \eta_1^{(N,L)}(t)}}$$

has asymptotically (for $L \gtrsim 50$) a standard normal distribution, as a consequence of the central limit theorem. Thus:

$$\text{Prob} \left\{ \left| \frac{\eta_1^{(N,L)}(t) - E\Xi^{(N)}(t)}{\sqrt{\text{Var} \, \eta_1^{(N,L)}(t)}} \right| \leq a_\alpha \right\} \sim \alpha, \quad \alpha \in (0, 1),$$

(3.27)

where the value of $a_\alpha$ is determined from statistical tables.

Note that:

$$\text{Var} \, \eta_1^{(N,L)}(t) = \frac{1}{L} \text{Var} \, \Xi^{(N)}(t) \sim \frac{1}{L} \eta_2^{(N,L)}(t).$$

(3.28)

A confidence interval can therefore be constructed as

$$I_\alpha = \left[ \eta_1^{(N,L)}(t) - a_\alpha \sqrt{\frac{\eta_2^{(N,L)}(t)}{L}}, \eta_1^{(N,L)}(t) + a_\alpha \sqrt{\frac{\eta_2^{(N,L)}(t)}{L}} \right],$$

(3.29)

where $\alpha$ is the confidence level. This means that

$$\text{Prob} \left\{ E\Xi^{(N)}(t) \in I_\alpha \right\} \sim \alpha.$$

(3.30)
Throughout this work, we use a confidence level of 99.9%, i.e., $\alpha = 0.999$, $a_\alpha = 3.29$, to construct our confidence intervals.
Chapter 4

Behaviour of the Stochastic Algorithm

4.1 Algorithm

We can now be confident that simulation of coagulation events and inflow/outflow events can each be performed accurately. Extension of stochastic simulation to include more than one process proceeds as follows. For each process (labelled \( \alpha \)) occurring, an associated waiting time parameter, \( \rho_\alpha(p) \) is derived. Then, the waiting time between events is exponentially distributed (Equation (2.7)) with parameter \( \rho(p) = \sum_\alpha \rho_\alpha(p) \). The event, \( \alpha \), occurring after this time is chosen according to the probabilities \( \frac{\rho_\alpha(p)}{\rho(p)} \), and the stochastic jump associated with this event is performed.

We can apply this approach to a situation common in the modelling of nanoparticle dynamics (see Chapter 5); simultaneous coagulation and particle inception:

\[
\frac{\partial}{\partial t} c(x,t) = \frac{1}{2} \sum_{y=1}^{x-1} K(x-y,y)c(x-y,t)c(y,t) - \sum_{y=1}^{\infty} K(x,y)c(x,t)c(y,t) + I(t)c_{in}(x). \tag{4.1}
\]

This equation is simply the Smoluchowski coagulation equation with a source term added. \( I(t) \) describes the rate of inception of particles (equivalent to the
factor $\frac{1}{\tau}$ in Equation (3.1)), while $c_{in}(x)$ describes their size distribution. Typically, in modelling of flame aerosol synthesis, the incepted particles are assumed to be monomers, i.e., the size distribution is monodisperse of unit size:

$$c_{in}(x) = \begin{cases} 1 & : x = 1, \\ 0 & : \text{otherwise.} \end{cases} \quad (4.2)$$

Combining the simulation methods for the processes of coagulation and particle inception results in the stochastic algorithm we use to simulate solutions to Equation (4.1) with coagulation kernel, $K(x, y)$, given by Equation (2.22) and majorant kernel, $\hat{K}(x, y)$, given by Equation (2.39). Organisation of the particle system is done according to Section 2.2.2.

1. Generate the initial state $P^N(x, 0) = p(x)$
2. Wait an exponentially distributed time step $\tau$ with parameter (c.f. (2.7))

$$\hat{\rho}(p) = \hat{\rho}_K(p) + \rho_{in}(p)$$

$$= \frac{\sqrt{2}}{N} \left( (n - 2) \sum_{i=1}^{n} x_i^{-\frac{3}{2}} + \sum_{i=1}^{n} x_i^{-\frac{3}{2}} \sum_{i=1}^{n} x_i^{-\frac{3}{2}} \right) + In_{in}.$$  

3. With probability

$$\frac{\rho_{in}(p)}{\hat{\rho}(p)},$$

go to step 4. Otherwise go to step 5.
4. Perform a source step, i.e.,

(a) Add a particle of size 1 to the system.
(b) Go to step 2.
5. Perform a coagulation step, i.e.,

(a) With probability (c.f. (2.41))

$$\frac{\hat{\rho}_1(p)}{\hat{\rho}_1(p) + \hat{\rho}_3(p)},$$

go to step 5b. Otherwise go to step 5c.
(b) Use $\hat{K}_1$ to generate indices $i$ and $j$, i.e.,

i. Generate $i$ according to:

A. Choose the group index, $z$, according to the probabilities:

$$P_z = \frac{1}{c} \sum_{k=1}^{\alpha_z} y_{z,k}^{\frac{2}{DF} - \frac{1}{2}}, \quad z = 1, \ldots, \psi,$$

where

$$c = \sum_{l=1}^{n} x_l^{\frac{2}{DF} - \frac{1}{2}} = \sum_{z=1}^{\psi} \sum_{k=1}^{\alpha_z} y_{z,k}^{\frac{2}{DF} - \frac{1}{2}}.$$

B. Choose the particle index $k = 1, \ldots, \alpha_z$ uniformly within the group $z$.

C. The particle index is accepted with probability

$$\frac{y_{z,k}^{\frac{2}{DF} - \frac{1}{2}}}{b_{z,k}^{\frac{2}{DF} - \frac{1}{2}}}.$$

Otherwise, go to step 5(b)iB.

ii. Generate $j$ uniformly on the set $i = 1, \ldots, n$.

iii. If $i = j$ return to step 5b. Otherwise go to step 5d.

(c) Use $\hat{K}_3$ to generate indices $i$ and $j$, i.e.,

i. Generate $i$ according to:

A. Choose the group index, $z$, according to the probabilities

$$P_z = \frac{1}{c} \sum_{k=1}^{\alpha_z} y_{z,k}^{\frac{2}{DF}}, \quad z = 1, \ldots, \psi,$$

where

$$c = \sum_{l=1}^{n} x_l^{\frac{2}{DF}} = \sum_{z=1}^{\psi} \sum_{k=1}^{\alpha_z} y_{z,k}^{\frac{2}{DF}}.$$

B. Choose the particle index $k = 1, \ldots, \alpha_z$ uniformly within the group $z$. 
C. The particle index is accepted with probability

\[ \frac{y_{z,k}^{\frac{1}{2}}}{b_{z,k}^{\frac{1}{2}}} \]

Otherwise, go to step 5(c)iB.

ii. Generate \( j \) according to:

A. Choose the group index, \( z \), according to the probabilities

\[ P_z = \frac{1}{c} \sum_{k=1}^{\alpha_z} y_{z,k}^{\frac{1}{2}}, \quad z = 1, \ldots, \psi, \]

where

\[ c = \sum_{l=1}^{n} x_l^{-\frac{1}{2}} = \sum_{z=1}^{\psi} \sum_{k=1}^{\alpha_z} y_{z,k}. \]

B. Choose the particle index \( k = 1, \ldots, \alpha_z \) uniformly within the group \( z \).

C. The particle index is accepted with probability

\[ \frac{y_{z,k}^{\frac{1}{2}}}{b_{z,k}^{\frac{1}{2}}} \]

Otherwise, go to step 5(c)iiB.

iii. If \( i = j \) return to step 5c. Otherwise go to step 5d

(d) With probability

\[ \frac{K(x_i, x_j)}{K(x_i, x_j)} \]

perform a coagulation jump, \( i.e., \) remove the particles \( x_i \) and \( x_j \) and add a particle of size \( x_i + x_j \). Otherwise, the interaction is fictitious, \( i.e., \) nothing changes.

(e) Go to step 2
4.2 Numerical Results

4.2.1 Sample Results

To study the efficiency of this stochastic algorithm we perform repeated simulations with varying values of \( N \) and calculate confidence intervals for the results we obtain. In all our simulations we take \( I = 0.5 \) and \( D_F = 2.1 \) and we set the initial condition to be

\[
e(0, x) = c_0(x) = \begin{cases} 
1 & \text{for } x = 1, \\
0 & \text{otherwise}.
\end{cases}
\]  

(4.3)

Figures 4.1 and 4.2 show typical results that can be obtained from the stochastic simulation. Figure 4.1 shows a histogram of the particle size distribution, with the lower confidence bound given by the solid line and the upper confidence bound given by the dotted line. The particle sizes are grouped in bins as described in Section 2.2.2 with a value of \( \beta = 2 \), in order to show the increasing concentration of the larger particles as time increases. It can also be seen that the concentration of smaller particles remains approximately constant as time increases, due to the formation of new monomer particles, as modelled by the source term.

Figure 4.2 shows the time evolution of three moments of the particle size distribution. As expected, the higher the moment, the more sensitive it is to the larger particle sizes, and therefore the wider the confidence bounds.

4.2.2 Convergence

To study the validity of our improved stochastic algorithm, it is useful to consider the convergence properties, i.e., as we increase \( N \), and therefore increase the required CPU times, how quickly does the simulated solution converge to the exact solution?

In calculating mean values and confidence intervals for our simulated functionals (see Section 3.4), we can also estimate the errors associated with these results. Recall we calculate an empirical mean value, \( \eta_1^{(N,L)}(t) \), to approximate the exact value of a functional, \( F(t) \). The error of this approximation is denoted as

\[
e^{(N,L)} = |\eta_1^{(N,L)}(t) - F(t)|,
\]  

(4.4)
Figure 4.1: Upper bound (dashed lines) and lower bound (solid lines) of the particle size distributions at various times.
Figure 4.2: Upper bound (dashed lines) and lower bound (solid lines) of the moments of the particle size distribution.
and consists of the following two components. The systematic error is the difference between the mathematical expectation of the random variable $\Xi^{(N)}(t)$ and the exact value of the functional, \(i.e.,\)

$$e_{\text{sys}}^{(N,L)}(t) = E\Xi^{(N)}(t) - F(t). \quad (4.5)$$

The statistical error is the difference between the empirical mean value and the expected value of the random variable, \(i.e.,\)

$$e_{\text{stat}}^{(N,L)}(t) = \eta_1^{(N,L)}(t) - E\Xi^{(N)}(t). \quad (4.6)$$

From calculating confidence intervals (Equation (3.29)) we have a probabilistic upper bound for the statistical error:

$$c_p^{(N,L)}(t) = a_\alpha \sqrt{\frac{\eta_2^{(N,L)}(t)}{L}}. \quad (4.7)$$

In order to describe the statistical error in \([0, t_{\text{tot}}]\) we split this time interval into \(M\) equidistant subintervals of length $\Delta t$ according to the discretisation

$$t_i = i\Delta t, \quad i = 0, 1, \ldots, M, \quad (4.8)$$

with $t_M = t_{\text{tot}}$ and use the quantity

$$c_{\text{stat}} = \max_i \{c_p^{(N,L)}(t)\} \quad (4.9)$$

as a measure for the statistical error.

To study the systematic error of the solution algorithm we use an approximation $\zeta(t)$ of the corresponding macroscopic quantity $F(t)$ obtained using a single run of the algorithm with as high a value of $N$ as is feasible. Here we use $N = 10^7$. Then the error

$$\tilde{e}^{(N,L)}(t) = |\eta_1^{(N,L)}(t) - \zeta(t)| \quad (4.10)$$

is a good approximation of the true error $e^{(N,L)}(t)$. In order to get an expression
for (4.10) on \([0, T]\) we calculate the quantity

\[
c_{\text{tot}} = \frac{1}{M+1} \sum_{i=0}^{M} \hat{e}^{(N,L)}(t_i)
\]

(4.11)
as an estimate for the average error in the time interval \([0, T]\).

The errors \(c_{\text{tot}}\) and \(c_{\text{stat}}\) are calculated for the average particle size, i.e., the ratio of the first moment of the particle size distribution to the zeroth. By using this term, we study a property both of the total number of particles and of the size of the particles in the population. Table 4.1 contains the results of the numerical study. In the table, \(t_{sr}\) is the CPU time (in seconds) needed for a single run. The simulations were all performed on a 866MHz Pentium PC.

**Table 4.1:** Computational study for improved stochastic algorithm \((N \times L = 6.4 \times 10^7)\).

<table>
<thead>
<tr>
<th>(N)</th>
<th>(c_{\text{stat}})</th>
<th>(c_{\text{tot}})</th>
<th>(t_{sr} [s])</th>
<th>(t_{sr}/N \times 10^4 [s])</th>
</tr>
</thead>
<tbody>
<tr>
<td>125</td>
<td>0.0773</td>
<td>2.13</td>
<td>0.012</td>
<td>0.94</td>
</tr>
<tr>
<td>250</td>
<td>0.0681</td>
<td>1.26</td>
<td>0.024</td>
<td>0.97</td>
</tr>
<tr>
<td>500</td>
<td>0.0633</td>
<td>0.697</td>
<td>0.048</td>
<td>0.96</td>
</tr>
<tr>
<td>1000</td>
<td>0.0607</td>
<td>0.368</td>
<td>0.097</td>
<td>0.97</td>
</tr>
<tr>
<td>2000</td>
<td>0.0601</td>
<td>0.180</td>
<td>0.22</td>
<td>1.1</td>
</tr>
<tr>
<td>4000</td>
<td>0.0601</td>
<td>0.0896</td>
<td>0.42</td>
<td>1.0</td>
</tr>
<tr>
<td>8000</td>
<td>0.0609</td>
<td>0.0423</td>
<td>0.92</td>
<td>1.2</td>
</tr>
<tr>
<td>16000</td>
<td>0.0616</td>
<td>0.0156</td>
<td>2.1</td>
<td>1.3</td>
</tr>
</tbody>
</table>

For a large enough number of runs, so that the systematic error is larger than the statistical error, we can estimate the order of convergence. Figure 4.3 shows \(c_{\text{tot}} \pm c_{\text{stat}}\) plotted against \(N\). The solid line indicates that the systematic error decreases as \(1/N\).

### 4.2.3 Efficiency

To give an idea of the efficiency of the improved stochastic algorithm, we compare it with a standard Monte Carlo algorithm; we use the algorithm proposed by Gillespie (1975) (see Appendix A), modified to include a source term. We also examine the efficiency of our new majorant kernel by comparison with the linear
Figure 4.3: Order of convergence for improved stochastic algorithm. The solid line shows the slope $\frac{1}{N}$.

majorant (2.25). Table 4.2 and Figure 4.4 show greatly improved efficiency of the new algorithm. It can be seen that for the Gillespie algorithm, the single-run CPU time, $t_{sr}$, increases as $N^2$. According to (2.8), the average number of time steps in a given time interval will be proportional to $N$, and due to the method of selecting $i$ and $j$, the CPU time to perform each coagulation step is proportional to $N$.

In contrast, we have a number of different cases, depending on the value of $N$, that our improved stochastic algorithm gives for the relationship between $t_{sr}$ and $N$. For low $N$, our algorithm gives $t_{sr}$ proportional to $N$. The number of time steps in a given time interval is still proportional to $N$, but the slowest step in the process is the calculation of the confidence intervals as described in Section 4.2.2. This process is not dependent on $N$, so the overall CPU time for a single run is proportional to $N$. It is only as $N$ increases that the rate of increase of $t_{sr}$ goes beyond linear. As $N$ increases, the next step of the algorithm to dominate is the bin selection step (step 5(b)iB, 5(c)iB or 5(c)iib in the simulation algorithm). As the number of bins to choose from increases as the logarithm of
the particle number (cf. (2.18) and (2.19)), the CPU time of each coagulation step will increase as $(\ln N)$. Finally, due to the necessity of reorganisation of the particle size array at each coagulation step, we reach the case where the CPU time for each coagulation step increases as $N$ and therefore $t_{sr}$ is proportional to $N^2$ (as in the case of the Gillespie algorithm). But this does not happen until the particle number is much greater than is necessary for a reasonably accurate simulation.

Table 4.2: Computational study for Gillespie algorithm ($N \times L = 8 \times 10^5$).

<table>
<thead>
<tr>
<th>$N$</th>
<th>$c_{stat}$</th>
<th>$c_{tot}$</th>
<th>$t_{sr}$ [s]</th>
<th>$t_{sr}/N \times 10^2$ [s]</th>
</tr>
</thead>
<tbody>
<tr>
<td>125</td>
<td>0.684</td>
<td>2.16</td>
<td>0.14</td>
<td>0.11</td>
</tr>
<tr>
<td>250</td>
<td>0.615</td>
<td>1.27</td>
<td>0.55</td>
<td>0.21</td>
</tr>
<tr>
<td>500</td>
<td>0.568</td>
<td>0.704</td>
<td>2.1</td>
<td>0.42</td>
</tr>
<tr>
<td>1000</td>
<td>0.525</td>
<td>0.399</td>
<td>8.3</td>
<td>0.83</td>
</tr>
<tr>
<td>2000</td>
<td>0.555</td>
<td>0.162</td>
<td>32</td>
<td>1.6</td>
</tr>
<tr>
<td>4000</td>
<td>0.533</td>
<td>0.118</td>
<td>130</td>
<td>3.2</td>
</tr>
<tr>
<td>8000</td>
<td>0.611</td>
<td>0.0566</td>
<td>497</td>
<td>6.2</td>
</tr>
<tr>
<td>16000</td>
<td>0.612</td>
<td>0.0459</td>
<td>1980</td>
<td>12</td>
</tr>
</tbody>
</table>

Figure 4.5 compares the single run CPU time (for constant $N$) as a function of $t_{sim}$, the time variable in the simulation, using our new majorant kernel and the linear majorant. Without the source term, the CPU time for both kernels levels off fairly soon, as by this point there are very few particles left in the system (with coagulation as the only process, the number of particles is strictly decreasing). When the source term is included, the CPU time is significantly increased. When our new majorant is used, CPU time increases linearly with $t_{sim}$. For the linear majorant, CPU time increases slightly faster than linearly. In both cases, our new majorant gives a reasonable ($10 - 50\%$) cut in CPU time as compared with the linear kernel, but the ratio of CPU times (linear to new) increases with increasing $t_{sim}$. This can be explained by considering the acceptance efficiency (2.29) of the two kernels. At later simulation times, the particle system contains larger particles. The ratio $\frac{t}{x_{min}}$ increases, and the efficiency of the simulation decreases. Figure 4.6 shows the relative number of fictitious jumps for both majorants, with and without a source term.

The new majorant kernel gives an increase in efficiency of simulation that is
more significant for longer simulation times. These results, for a constant particle number, $N$, can be considered to hold for other particle numbers by examining Figure 4.7, which shows CPU time increasingly linearly with particle number. Thus, we can expect a similar gain in efficiency regardless of particle number.

The order of convergence for the Gillespie algorithm is compared with $\frac{1}{N}$ in Figure 4.8. Due to the poor efficiency of this method it is not feasible to perform sufficient runs such that the systematic error is significantly larger than the statistical error. However, the data do seem to fit the $\frac{1}{N}$ line plotted on the figure. This is to be expected, as the convergence of $P^N(x,t)$ does not depend on any majorant that is used (or not used).

We have developed and studied an improved stochastic simulation algorithm to model the nano-particle dynamics applicable to, for example, the production of silica by flame aerosol synthesis (cf. Equations (5.1)–(5.6)). The coagulation was simulated according to Eibeck and Wagner (2000b), with the introduction of a new majorant kernel. The new majorant kernel allows us to study a wider range of problems than the linear majorant, previously suggested by Eibeck and

![Figure 4.4: CPU time for a single run for the Gillespie algorithm (solid) and our improved algorithm (dotted).]
Figure 4.5: CPU time for a single run for the linear majorant and for our new majorant. $t_{\text{sim}}$ is the simulation time.
Figure 4.6: Comparison of the efficiency of the two majorants.

Figure 4.7: CPU time for a single run for the linear majorant and our new majorant. \(N\) is the particle number.
Wagner (2001). A source term was included in the population balance equation and was also simulated.

The results can easily be presented in the form of particle size distributions (PSDs) or any function of the PSDs, such as moments.

The convergence properties of the algorithm were studied, to give an indication of the decrease of the systematic error when the particle number, \(N\), is increased. For a very large number of runs, the statistical error is smaller than the systematic error, and the systematic error decreases as \(\frac{1}{N}\). It is likely however, that for normal simulations, the number of runs will be sufficiently small that the systematic error is smaller than the statistical error and hence the exact solution will lie within the calculated confidence bounds.

The simulation algorithm was compared to a standard Monte Carlo algorithm (i.e., one not using fictitious jumps) to determine the improvement in efficiency. We also compared the efficiency of our new majorant kernel with that of the linear kernel.

By comparing it to the algorithm proposed by Gillespie (1975) (see Appendix A) we see that we have reduced CPU times by orders of magnitude, and made it possible to achieve accuracy that the inefficiency of the Gillespie algorithm prohibited. We also achieve good gains in efficiency (up to 50\% cut in CPU time) by using our new majorant kernel in place of the linear majorant, while also being able to study a wider range of problems. This is very promising for future work, both in the study of fumed silica and in other fields, such as soot formation, where the population balance equation is used.

Appendices

A Gillespie Algorithm

When using the stochastic solution algorithm as proposed by Gillespie (1975), there is no majorant kernel, and therefore the calculation of the time increment requires a double sum of the coagulation kernel over \(i\) and \(j\) (cf. (2.8)). The most efficient way of updating this double sum after each time step is to store for each
value of $i$ the function:

$$C_i = \sum_{j=i+1}^{n} K(x_i, x_j) \quad i = 1, \ldots, n - 1.$$  \hfill (4.12)

These functions can be updated at each time step, in conjunction with a summation over $i$ to give the function

$$C_0 = \sum_{i=1}^{n-1} C_i = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} K(x_i, x_j),$$  \hfill (4.13)

which is used in the calculation of the time increment $\tau$. Thus, each time step requires a number of calculations that is of the order of $n$. (The notation $C_i$ and $C_0$ follows the notation used by Gillespie (1975).)

The algorithm as applied to the problem of silica formation and coagulation is given as follows.

1. Generate the initial state $P^N(0) = p(x)$

2. Wait an exponentially distributed time step $\tau$ with parameter (c.f. Equations 2.8, 4.12)

$$\rho(p) = \rho_K(p) + \rho_{in}(p) = \frac{1}{N} \sum_{i=1}^{n-1} C_i + In_{in}$$

3. With probability

$$\frac{\rho_{in}(p)}{\rho(p)}$$

go to step 4. Otherwise go to step 5.

4. Perform a source step, i.e.

(a) Add a particle of size 1 to the system.

(b) Update the stored values of $C_i$.

(c) Go to step 2.

5. Perform a coagulation step, i.e.
(a) Uniformly generate index \( i \) on the domain 1, 2, \ldots, \( n \) and index \( j \) on the domain 1, \ldots, \( i - 1, i + 1, \ldots, n \).

(b) With probability

\[
\frac{K(x_i, x_j)}{\max_{i,j} K(x_i, x_j)}
\]

accept the indices \( i \) and \( j \), otherwise return to step 5.

(c) Remove the particles \( x_i \) and \( x_j \) and add a particle of size \( x_i + x_j \).

(d) Update the stored values of \( C_i \).

(e) Go to step 2.
Figure 4.8: Order of convergence for the Gillespie algorithm. The solid line shows the slope $\frac{1}{N}$. 
Chapter 5

Nanoparticle Synthesis in Flames

5.1 Experimental Comparison

We have developed an improved, efficient stochastic algorithm suitable for simulating solutions to population balance problems typical to the field of nanoparticle dynamics. We now wish to demonstrate its suitability by comparing the results with experimental studies and with established solution techniques. Further, we wish to demonstrate some of the features that make stochastic solution methods attractive in comparison to other techniques.

Nano structured materials have drawn attention in recent years. Reducing the size of materials down to a nanometer scale leads to a significant change of their physical and chemical properties. These are the subject of extensive research ranging from fundamental investigations on the chemical and physical qualities of such materials to optimising engineering routes of nanoparticle synthesis. An established way of producing the smallest particles is by aerosol flame reactors, which allow large scale production of materials such as carbon black or fumed silica. This work focuses on the formation of fumed silica by gas phase synthesis. For optimising purposes a thorough understanding of the process is crucial and there is a demand for models that incorporate the complex gasphase reactions of the combustion process and particle dynamics in the reactor environment, which are generally characterized by small residence times.

Flame aerosol technology is an established industrial process, but the fundamentals of the process are not yet well understood. It is a difficult process to study due to the fact that chemical reaction and particle growth take place
extremely fast in typical process conditions. The characteristic times for reaction and particle growth (under a second) are smaller than the typical residence time in the reactor, and it is this, coupled with high process temperatures, that makes it difficult to collect representative samples for particle characterisation and model development.

In this section we compare the results of the algorithm described in Chapter 4 to experimental and numerical results published by Lindackers et al. (1997). The tested system is a low pressure flame reactor in which an Ar diluted H\textsubscript{2}/O\textsubscript{2} flame was doped with a SiH\textsubscript{4} precursor, producing SiO\textsubscript{2} molecules that subsequently coagulate to form large particle clusters.

5.2 Model

An accurate description of the flame structure is important for the prediction of particle formation. Therefore the gasphase reactions were described by a combined mechanism, consisting of a detailed H\textsubscript{2}/O\textsubscript{2}/Ar- and a skeletal SiH\textsubscript{4}/O\textsubscript{2} reaction scheme (Lindackers et al., 1997). The species profiles for the laminar premixed flame in steady state were computed using the PREMIX code (Kee et al., 1985). See Grosschmidt et al. (2003) for further details.

In the nucleation process new particles are formed from matter that is not in a particulate state (Hounslow, 1998). In our test case silica powder is generated from the precursor SiH\textsubscript{4} by high temperature reactions. The common approach to describe the transition from gaseous to condensed solid state is given by e.g. Ulrich (1971); Pratsinis (1998); Johannessen et al. (2000); Lindackers et al. (1997). The combustion product SiO\textsubscript{2} is considered to become the smallest stable member of the particle representation (Ulrich, 1971). Hence the rate of particle formation is directly coupled to the gas phase production rates of SiO\textsubscript{2}.

The specific form of the population balance that we examine in this section is the discrete form, with coagulation and a source term being the processes of interest:
\[ \frac{\partial}{\partial t} c(x, t) = \frac{1}{2} \sum_{y=1}^{x-1} K(x - y, y) c(x - y, t) c(y, t) - \sum_{y=1}^{\infty} K(x, y) c(x, t) c(y, t) + I(t) c_{in}(x), \quad (5.1) \]

with initial condition
\[ c(x, 0) = c_0(x) \geq 0 \quad (5.2) \]

The coagulation kernel has different forms depending on which regime the system is in (Friedlander, 1977). The regimes are characterised by the Knudsen number, \( Kn = 2\lambda_f/d \). In the continuum regime (\( Kn \leq 0.1 \)), the coagulation kernel is modelled by assuming a Stokes’ law drag coefficient, and \( K(x_i, x_j) \) is given by:
\[ K(x_i, x_j) = \frac{2k_BT}{3\mu} \left( \frac{1}{x_i^{1/3}} + \frac{1}{x_j^{1/3}} \right) \left( x_i^{1/3} + x_j^{1/3} \right) \quad (5.3) \]

In the slip flow regime (0.1 \( \leq Kn \leq 1 \)) the kernel is modified slightly (Kazakov and Frenklach, 1998):
\[ K(x_i, x_j) = \frac{2k_BT}{3\mu} \left( 1 + 1.257Kn_i x_i^{1/3} + 1 + 1.257Kn_j x_j^{1/3} \right) \left( x_i^{1/3} + x_j^{1/3} \right) \quad (5.4) \]

In the free-molecular regime (\( Kn \geq 10 \)), \( K(x_i, x_j) \) is given by (Huang et al., 1991):
\[ K(x_i, x_j) = \left( \frac{3}{4\pi} \right)^2 \left( \frac{8\pi kT}{\rho} \right)^{1/2} \left( \frac{m_1}{\rho} \right)^{1/2} \left( \frac{1}{x_i} + \frac{1}{x_j} \right)^{1/2} \left( \frac{1}{D_{F_i}} + \frac{1}{D_{F_j}} \right) \quad (5.5) \]

In the transition regime (1 \( \leq Kn \leq 10 \)), the coagulation kernel is assumed to be twice the harmonic mean of the slip flow kernel and the free molecular kernel (Pratsinis, 1988).

For this study, we apply the free-molecular regime kernel, with \( D_F = 2.1 \) (Lindackers et al., 1997).

In this case, the source term refers to monomers \((i.e., \) single molecules of the
substance of interest), so the inflowing particle size distribution is:

\[
c_{in}(x) = \begin{cases} 
1 : & x = 1, \\
0 : & \text{otherwise}.
\end{cases}
\]  

(5.6)

## 5.3 Results

The calculated gas-phase species profiles are shown in Figure 5.1. This enables the source rate, \( I \), to be calculated as a function of distance along the burner. The relationship between distance and time was determined by employing a Lagrangian view of particles in a reference volume that propagates along the flame front. The production rate of \( \text{SiO}_2 \) exhibits a significant peak early in the reaction zone, which means that the particle formation process is located in the zone near the flame front. The post flame zone and the areas downstream of the chamber are governed only by coagulation processes.

![Figure 5.1: Molefraction profiles of major gas-phase species and production rate of SiO\textsubscript{2} (Grossschmidt et al., 2003).](image)

The evolution of the mean particle mass is shown in Fig. 5.2 using the method of moments (e.g. Frenklach, 1985; Frenklach and Harris, 1986; Grossschmidt et al., 2003) (dashed line) and stochastic simulation (solid lines). The results for both
methods correspond very well to the numerical investigations in Lindackers et al. (1997). The discrepancy between the simulations and the measured data is explained by the fact that the particles were ionized during the experiments, which increased the coagulation probability. This is thoroughly discussed by Lindackers et al. (1997).

![Figure 5.2: Mean particle mass as a function of the height above the burner for the method of moments and stochastic simulation. The discrepancy between simulation and measurements is explained by the fact that the ionization of the particles in the experiments are not accounted for in the models.](image)

5.3.1 Particle Size Distributions

Stochastic simulation, although slower than the method of moments, has the benefit of being able to provide a complete statistical description of the particle population at any point in time. This can be used to obtain a very accurate approximation of the solution of the population balance equation. It is common, while trying to solve the population balance equation, to use an assumed shape probability density function for the particle size distribution. These assumed shapes include monodisperse (Panda and Pratsinis, 1995) and lognormal
(Whitby, 1981), and this second case is the one we examine; we try to find a fit for the distributions obtained from stochastic simulation of the test case described above.

The lognormal distribution is given as:

\[ f_X(x) = \frac{1}{B x \sqrt{2\pi}} \exp \left( -\frac{1}{2} \left( \frac{\ln x - A}{B} \right)^2 \right) \] (5.7)

where \( A \) and \( B \) are parameters describing the location and scale of the distribution. From this distribution, the following relationships (among others) can be obtained:

\[
E(X) = \exp \left( A + \frac{B^2}{2} \right) \] (5.8)

\[
\text{var}(X) = \exp (2A + B^2) \left( \exp (B^2) - 1 \right) \] (5.9)

\[
\text{Mode} = \exp (A - B^2) \] (5.10)

and one has a choice of which two characteristics of the distribution to use to select the parameters \( A \) and \( B \) in order to obtain a fit for the particle size distribution.

Taking the first two moments of the distribution (easily obtainable by using the method of moments) to determine \( A \) and \( B \) from the expectation and the variance results in an assumed shape probability density function that does not match up well with the particle size distributions obtained from stochastic simulation (see Figure 5.3).

An alternative method is to use the mean and the mode of the distribution. This results in a much better fit, but requires knowledge of the location of the peak of the distribution, which is not available experimentally or from the method of moments. However, it can be observed from applying this method at various values of \( t \) that the parameter \( B \) remains approximately constant, in our case at a value of 1.37. Thus, there is one remaining parameter to be determined, which can be estimated from the expectation of the particle size distribution, which in turn can be obtained from experiments or from the method of moments.

In Figure 5.4, the particle size distribution obtained from stochastic simulation is compared with the assumed shape lognormal probability density function using:

\[
B = 1.37, \quad A = \ln (E(X)) - \frac{B^2}{2} \] (5.11)
Figure 5.3: Comparison of the particle size distribution obtained from stochastic simulation (solid lines) with an assumed shape lognormal probability density function (dashed line). The parameters for the assumed shape probability density function were obtained from the expectation and variance of the distribution as $A = 5.818$, $B = 0.996$. 
This method even predicts closely the form of the particle size distribution at early times, when the lognormal form is not yet apparent and a peak is not observed. In this case, the lognormal distribution with the parameters obtained as above needs to be renormalised to account for the fact that we do not encounter any particles of size less than one. (See Figure 5.4(a))

At the exit of the burner (distance co-ordinate = 90mm) the comparison of simulated and assumed shape particle size distributions is shown in Figure 5.4(d). Good agreement is still obtained.

Figure 5.4: Comparison of stochastic simulation (solid lines) and assumed shape lognormal fit (dashed lines) for different heights above the burner
It should be noted that for very early times, when particle inception is still occurring, this method does not give such a good fit; indeed it seems only to work in the case where coagulation is the only mechanism occurring. This numerical observation of a self preserving lognormal distribution provides further evidence of a phenomenon that has long been observed for pure coagulation processes but that has not been fully explained (see Aldous (1999) for a mathematical examination).
Chapter 6

Simulation of Coalescence and Breakage

6.1 Breakage Process

Mathematical consideration of the coagulation process is well established (von Smoluchowski, 1916), and the particle size evolution in the continuous case is given as:

\[
\frac{\partial c(x,t)}{\partial t} = \frac{1}{2} \int_0^x K(x-y,y)c(x-y,t)c(y,t)dy - \int_0^\infty K(x,y)c(x,t)c(y,t)dy.
\] (6.1)

In the continuous case, we now write \(c(x,t)dx\) as the number of particles (per unit volume) in the size range \([x, x + dx]\), i.e., the units of \(c(x,t)\) are \(m^{-6}\). The meaning and units of \(K(x,y)\) are unchanged.

The mathematical treatment of binary breakage is less well established, with contradictory formulations proposed in the literature. There are two ways to consider the breakage characteristics of a population of particles. One is to describe the breakage by a breakage frequency, \(g(x)\), and a probability density function for the sizes of the daughter particles, \(\beta(x,y)\). \(g(x)\) is the frequency at which a particle of size \(x\) undergoes breakage and \(\beta(x,y)dy\) gives the probability that the size of a particle formed from breakage of a particle of size \(x\) is in the range \([y, y + dy]\). The evolution of \(c(x,t)\) is given by:

\[
\frac{\partial c(x,t)}{\partial t} = -c(x,t)g(x) + 2 \int_x^\infty c(y,t)g(y)\beta(y,x)dy.
\] (6.2)
This representation is commonly used in the literature (e.g. Coulaloglou and Tavlarides, 1977) when it is required to match experimental observations with theoretical predictions of breakage rates. The breakage rate, $g(x)$, is measurable, although it can be harder to do so for the daughter particle size distribution function. An assumed distribution is sometimes used instead of a theoretical model, e.g. uniform (Narsimhan et al., 1979), Beta (Haverland et al., 1987) or normal (Coulaloglou and Tavlarides, 1977).

The second way of looking at the breakage rate is to take the frequency of production of the daughter particles, so that the function $f(x, y)$ is the rate at which a particle of size $x$ and a particle of size $y$ are formed from the breakage of a particle of size $x + y$. This representation can sometimes be more convenient when an analytical solution to the population balance equation is being sought. It should be noted here that there is some disagreement in the literature as to the exact form of the population balance equation when $f(x, y)$ is used. Some authors (e.g. Ziff and McGrady, 1985) write the evolution of the particle size distribution as:

$$\frac{\partial c(x,t)}{\partial t} = -c(x,t) \int_0^x f(y, x - y)dy + 2 \int_x^\infty c(y,t)f(x, y - x)dy, \quad (6.3)$$

while others (e.g. Eibeck and Wagner, 2000a) write:

$$\frac{\partial c(x,t)}{\partial t} = -\frac{1}{2}c(x,t) \int_0^x f(y, x - y)dy + \int_x^\infty c(y,t)f(x, y - x)dy. \quad (6.4)$$

The discrepancy between these two equations results from confusion over the exact relationship between $f(x, y)$, $g(x)$ and $\beta(x, y)$, and whether double counting is taken into account. The two different equations arise from using either:

$$f(y, x - y) = g(x)\beta(x, y), \quad (6.5)$$

or

$$f(y, x - y) = 2g(x)\beta(x, y), \quad (6.6)$$

to describe the formation rate of the daughter particles.

It is perhaps easiest to formulate the correct relationship using a discrete example; that of depolymerization, as described by Ziff and McGrady (1985).
Consider a polymer chain of \( k \) units. The chain has \( k - 1 \) bonds that could each be broken on a binary breakage step. Assuming a uniformly distributed daughter chain length distribution, we can write \( \beta(k, j) = \frac{1}{k-1} \). If, for simplicity, we set the breakage rate of this chain as \( g(k) = (k - 1) \text{s}^{-1} \) then we can work out the production rate of the daughter chains, \( f(k - j, j) \): on average, in one second, we would expect one breakage to occur at each of the \( k - 1 \) available breakage sites. Thus, in one second, we produce two chains of length one, two chains of length two and so on. We can write \( f(k, j) = 2 \text{s}^{-1} \) for \( j = 1, 2, \ldots, k - 1 \), which is clearly twice the product of \( g(k) \) and \( \beta(k, j) \), thus Equation (6.6) is the correct one.

In general, we can write the relationship between \( f(x, y), g(x) \) and \( \beta(x, y) \) in the continuous case as:

\[
f(y, x - y) = 2g(x)\beta(x, y) \\
\int_0^x f(y, x - y)dy = 2 \int_0^x g(x)\beta(x, y)dy \\
= 2g(x) \int_0^x \beta(x, y)dy = 2g(x).
\] (6.7)

Here we have made use of the fact that \( \beta(x, y) \) is a density distribution normalised on the interval \([0, x]\).

Combining the two processes, coalescence and breakage, gives us the population balance equation that we wish to solve:

\[
\frac{\partial c(x, t)}{\partial t} = \frac{1}{2} \int_0^x K(x - y, y)c(x - y, t)c(y, t)dy - \int_0^\infty K(x, y)c(x, t)c(y, t)dy \\
- c(x, t)g(x) + 2 \int_x^\infty c(y, t)g(y)\beta(y, x)dy.
\] (6.8)

### 6.2 Direct Simulation Algorithm

The general method for direct simulation of solutions to the coalescence-breakage equation has been given by Eibeck and Wagner (2000a). Their paper concentrates on proving existence of a solution, and proposes a simulation algorithm for the coalescence-breakage case. This method can be described as the direct simulation algorithm (DSA) because a real particle ensemble is approximated by a stochastic
particle system, where one stochastic particle approximates a certain number concentration of real particles of that size.

We do not go into the details of the derivation of this method here, but refer to Eibeck and Wagner (2000a) and quote the algorithm.

1. Generate the initial state of the system \( P^N(x, 0) = p(x) \).

2. Wait an exponentially distributed time step \( \tau \) with parameter

\[
\rho(p) = \rho_K(p) + \rho_g(p) = \frac{1}{2N} \sum_{1 \leq i \neq j \leq n} K(x_i, x_j) + \sum_{i=1}^{n} g(x_i),
\]

i.e., the waiting time, \( \tau \) is chosen according to:

\[
\text{Prob}\{\tau(p) \geq s\} = \exp(-\rho(p) \cdot s), \quad s \geq 0.
\]

3. With probability:

\[
\frac{\rho_K(p)}{\rho_K(p) + \rho_g(p)},
\]

choose coalescence and hence go to step 4. Otherwise, choose breakage and go to step 5.

4. Perform a coalescence jump, i.e.,

   (a) Choose a pair \( i, j \) according to the index distribution

\[
\frac{K(x_i, x_j)}{2N\rho_K(p)}, \quad 1 \leq i \neq j \leq n.
\]

   (b) Remove the particles \( x_i \) and \( x_j \) and add a particle of size \( x_i + x_j \).

   (c) Go to step 2.

5. Perform a breakage jump, i.e.,

   (a) Choose an index \( i \) according to the distribution

\[
\frac{g(x_i)}{\rho_g(p)}, \quad 1 \leq i \leq n.
\]
(b) Choose a breakage part $y$ according to the distribution
\[ \beta(x_i, y), \]
and remove the particle $x_i$ and add particles of sizes $y$ and $x_i - y$.

(c) Go to step 2.

Note that for the purposes of this work we have simplified the Eibeck and Wagner DSA algorithm. The test case that we introduce in section 6.6 is simple enough that no majorant kernels (Eibeck and Wagner, 2000a,b) need be introduced. Extension of the algorithm above to include majorant kernels and fictitious jumps is, however, straightforward. For mathematical rigour, we should also include a number truncation parameter, $c_N$. As the breakage process results in an increase in number of particles, an upper limit must be set on this number to ensure that the system remains within the solution space defined in the existence proof. For all practical situations however, the particle number tends to a finite steady state value, so we can, by effectively setting this truncation parameter arbitrarily large, ignore its effects.

### 6.3 Mass Flow Algorithm

Eibeck and Wagner (2001) introduced the so-called Mass Flow Algorithm for stochastic solution of the Smoluchowski coagulation equation. In contrast to direct simulation, where one stochastic particle can be considered to represent one real particle in a theoretical sample space, the mass flow case uses one stochastic particle to represent mass concentration rather than number concentration.

More specifically, if $P^N(x, t)$ is a measure valued solution of the population balance equation, then in the mass flow algorithm, we seek the solution:

\[ Q^N(x, t) = xP^N(x, t), \quad (6.9) \]

where $Q^N(x, t)$ is the sequence of jump processes representing the behaviour of a system of particles whereby one stochastic particle represents a certain mass (as opposed to number) concentration. We use $q(x)$ to represent any realisation of this sequence of random variables. Now, in approximating a functional of the
concentration, \( \int_0^\infty \phi(x)c(x)dx \), we use:

\[
\int_0^\infty \phi(x)c(x)dx \sim \int_0^\infty \frac{\phi(x)}{x} \frac{1}{N} \sum_{i=1}^n \delta(x-x_i)dx \\
\sim \frac{1}{N} \sum_{i=1}^n \frac{\phi(x_i)}{x_i}.
\] (6.10)

Jourdain (2003) applied the mass flow algorithm to the case of fragmentation and coagulation in the discrete case. We now wish to satisfy ourselves that the mass flow algorithm is applicable to the breakage process in the continuous case. Following Wagner (2003), we can write for the breakage case the weak integral version of the breakage equation:

\[
\frac{d}{dt} \int_0^\infty \phi(x)Q^N(x,t)dx = \int_0^\infty \int_0^\infty \left[ \phi(y) - \phi(x) \right] \frac{2y}{x} g(x) \beta(x,y) dy Q^N(x,t)dx.
\] (6.11)

This leads to writing the breakage infinitesimal generator:

\[
\mathcal{A}_q^N \Phi(q) = \sum_{i=1}^n \int_0^\infty \left[ \Phi(J_g(q,i,y)) - \Phi(q) \right] \frac{2y}{x_i} g(x_i) \beta(x_i,y) dy,
\] (6.12)

where the breakage operator is given by:

\[
J_g(q,i,y) = q + \frac{1}{N} (\delta(x-y) - \delta(x-x_i)).
\] (6.13)

We note here that the symmetrical nature of \( \beta(x,y) \) means that:

\[
\int_0^x \phi(y) \beta(x,y) dy = \int_0^x \phi(x-y) \beta(x,x-y) dy,
\] (6.14)

and that:

\[
\beta(x,y) = 0 \quad \text{for} \; x < y.
\] (6.15)
CHAPTER 6. COALESCENCE AND BREAKAGE

Thus, in constructing a simulation algorithm, we make use of the fact that:

\[
\int_0^\infty y\beta(x,y)dy = \int_0^x y\beta(x,y)dy = \int_0^x (x-y)\beta(x,y)dy = \frac{1}{2} \int_0^x x\beta(x,y)dy = \frac{x}{2}.
\] (6.16)

We now expect to use as the distribution for selecting a breakage part, \(y\), (as opposed to using \(\beta(x,y)\) in the algorithm described above):

\[
\frac{y\beta(x,y)}{\int y\beta(x,y)dy} = \frac{y\beta(x,y)}{\frac{x}{2}} = \frac{2y\beta(x,y)}{x}.
\] (6.17)

This suggests the following simulation algorithm:

1. Generate the initial state of the system \(Q_N(x,0) = q(x)\).

2. Wait an exponentially distributed time step \(\tau\) with parameter

\[
\rho(q) = \rho_K(q) + \rho_g(q) = \frac{1}{N} \sum_{i,j=1}^{n} \frac{K(x_i,x_j)}{x_j} + \sum_{i=1}^{n} \int_0^{x_i} \frac{2y}{x_i} g(x_i)\beta(x_i,y)dy
\]

\[
= \frac{1}{N} \sum_{i,j=1}^{n} \frac{K(x_i,x_j)}{x_j} + \sum_{i=1}^{n} g(x_i),
\]

\(i.e.,\) the waiting time, \(\tau\) is chosen according to:

\[
\text{Prob}\{\tau(q) \geq s\} = \exp(-\rho(q) \cdot s), \quad s \geq 0.
\]

3. With probability:

\[
\frac{\rho_K(q)}{\rho_K(q) + \rho_g(q)},
\]

choose coalescence and hence go to step 4. Otherwise, choose breakage and go to step 5.

4. Perform a coalescence jump, \(i.e.,\)
(a) Choose a pair \(i, j\) according to the index distribution
\[
\frac{K(x_i, x_j)}{N\rho_K(q)x_j}, \quad 1 \leq i, j \leq n.
\]

(b) Remove the particle \(x_i\) and replace it with a particle of size \(x_i + x_j\).

(c) Go to step 2.

5. Perform a breakage jump, i.e.,

(a) Choose an index, \(i\), according to the distribution
\[
\frac{g(x_i)}{\sum_{i=1}^n g(x_i)}.
\]

(b) Choose a breakage part \(y\) according to the distribution
\[
\frac{2y\beta(x_i, y)}{x_i},
\]
and replace the particle of size \(x_i\) with one of size \(y\).

(c) Go to step 2

Note that, in a similar fashion to the algorithm presented in section 6.2, we have simplified this algorithm. In their original presentation of the mass flow algorithm, Eibeck and Wagner (2001) include a maximum size truncation parameter, \(b_N\). Similar to the particle number truncation parameter, \(c_N\), this parameter exists to prevent any individual particle growing too large and thus growing larger than the solution space defined in the existence proof. In coagulation only problems, where the particles grow larger without limit, this parameter can be useful in predicting gelation properties, but in general for coalescence-breakage problems, we can arbitrarily set \(b_N\) large enough that it does not affect our simulation.

6.4 Data Storage

The method used to store the stochastic particle sizes and associated functionals has a profound influence on the efficiency of generation of required probability
distributions. Often there is a trade-off between economy of storage and efficiency of generation, but we generally value low CPU times over low storage requirements.

### 6.4.1 Bins

Arranging the stochastic particles by size into bins is a method favoured by Eibeck and Wagner (2000b, 2001) for the case of coagulation only (see Section 2.2.2). The result is that, for \( n \) particles, the time taken for the index generation step is proportional to \( \log(n) \). However, after each stochastic jump, the bins must be updated. The process in which an empty space is removed from an array to reflect a particle being removed from that bin takes time of order \( n \). In this case, for large particle numbers, the overall process can take time of order \( n^2 \) (Goodson and Kraft, 2002).

A further problem encountered when the bin method is used is a consideration of the range of bin sizes. Sensible selection of \( \beta \) is essential to optimise CPU time, but this often results in larger storage requirements. Furthermore, for the coagulation-only case, there is a well-defined minimum particle size (the smallest particle in the initial distribution). This minimum particle size might not exist in a breakage problem, in which case a lowest bin must be set, which then contains all particles smaller than some cut-off size. The selection of \( \beta \) and the minimum bin-size in order to minimise the CPU time of the simulation is an extra and undesirable problem and so we look for a system of storage/index generation that is more appropriate for the coalescence-breakage process.

### 6.4.2 Binary Search

A binary search method of generating distributions is particularly suitable for use within the mass flow algorithm. The number of stochastic particles remains constant (unless a particle grows larger than \( b_N \), but this is rare, especially if \( b_N \) is chosen large enough), so if a binary tree is constructed, it remains the same height throughout the simulation.

Figure 6.1 shows an example of the construction of a binary tree. For simplicity, it is of height 3, so the particle number is \( 2^3 = 8 \). We are interested in
generating an index, $i$, according to the distribution:

$$\frac{\phi(x_i)}{\sum_{i=1}^{n} \phi(x_i)}.$$ \hspace{1cm} (6.18)

The function $\phi(x)$ is often simply the value $x$ raised to some power, but it need not be restricted to such cases.

Note that in Figure 6.1 the particles (on the lowest row of the tree) are labelled $i = 0, 1, \ldots, 7$, i.e., from 0 to $n - 1$, so from now on in this section, the numbering of particle indices will reflect this.

To choose a particle index, $i = 0, 1, \ldots, 7$, we generate a uniform random deviate on [0,1] and multiply it by $\sum_{i=0}^{7} \phi(x_i)$. We now check if this quantity is greater than the value of node 1 on the tree. If it is, we move down the tree to the right (1). If not, we move down to the tree to the left (0). If we move left, we repeat the process. If we move right, then we need to update our comparison quantity, by subtracting the value of node 1. Then we repeat the process down to the bottom. Note that by recording the value of 0 or 1 each time we move down a level, we end up with a binary representation of the particle index we are interested in (hence the renumbering of the particles). Note also that this binary representation shows which nodes on the tree need to be updated after a stochastic jump. Whenever a 0 is recorded as we move down the tree, the node that has just been passed will need updating.

The index generation and system update steps both now take CPU time of order $\log(n)$, so the overall CPU time required increases as $n \log(n)$ regardless of particle number.

A comparison of CPU time for the two methods (when applied to a simple test case, (see Section 6.6)) is shown in Figure 6.2. It can be seen that, despite similar runtimes for low particle numbers, the CPU time for the bins method increases slightly faster, and the two methods diverge at around $n = 10^4$. Even for particle numbers up to $10^6$, the CPU time for the binary search method increases just faster than linearly with particle number.
Figure 6.1: A binary tree, of height 3.

Figure 6.2: Single run CPU time comparison for the bins and binary methods.
6.5 Constant Number Stochastic Simulation

Coagulation and breakage are both mass-conserving processes, thus when using the mass flow algorithm, the number of stochastic particles remains constant when either of these processes is simulated. This is not the case when using the direct simulation algorithm. In a pure coagulation problem, simulated by DSA, the number of stochastic particles is strictly decreasing. This leads to problems if long simulation times are required, as the number of particles can decrease to the point where the increased variance is unacceptable, or even to the point where there is only one particle left. Conversely, in a pure breakage problem, the number of particles is strictly increasing, and there can be problems with increasing simulation time for the increased number of particles. There are also benefits to keeping the number of particles in our stochastic system constant purely for ease of bookkeeping: a constant number of particles means that the binary tree used for generating probability distributions is of a constant height, and no rearrangement of this structure is needed.

6.5.1 Updating the Particle System

The constant number stochastic simulation method is based on the ideas of Matsoukas and co-workers (Smith and Matsoukas, 1998; Lee and Matsoukas, 2000; Lin et al., 2002), whereby after every stochastic simulation step, the particle system is updated to maintain a constant number of stochastic particles. The approach originated with the particle doubling method of Sabelfeld et al. (1996). This method for simulating a pure coagulation problem involved three particle arrays that each behave according to the coagulation rate, decreasing in size. When the number of particles in each array is half its original amount, the arrays are combined to give three arrays of the original size again, and simulation proceeds as before. Matsoukas’ method involves updating the particle system at every step, by choosing a particle at random from the array and duplicating it to fill the gap that has been left by the coagulation event that has just occurred. This process is shown schematically in Figure 6.3.

In fact, for ease of simulation, this process can be combined into one step, where the first of the coagulating particles is replaced by the resulting coagulated particle, and the second is replaced by the duplicated particle. This process is
CHAPTER 6. COALESCENCE AND BREAKAGE

Figure 6.3: Schematic representation of a particle duplication step following a coagulation event.

shown in Figure 6.4. This process is also used for any stochastic jump where the number of particles decreases by one, e.g. an outflow step.

Figure 6.4: Schematic representation of a combined coagulation event and particle duplication step.

In the case of breakage (or any process where the particle number increases by one), one particle has to be removed. Again, this particle is chosen uniformly from the particle array. This process (condensed into one step) is shown in Figure 6.5.

6.5.2 Renormalisation

Following a particle duplication step (after coagulation) or a particle removal step (after breakage), we must renormalise the particle system, so that the simulation
remains an accurate approximation of the solution we seek. By renormalisation, we mean updating the value of the normalisation parameter $N$. The approach used here is slightly different to that of Matsoukas, in that the particle system updating can be considered to take place independently of the stochastic jumps themselves. Consider a stochastic particle system containing $n$ stochastic particles. After (for example) a coagulation step, the system contains $n - 1$ particles, with various functionals that are approximated by:

$$\Phi(t) = \int_0^\infty \phi(x)c(x,t)dx \sim \frac{1}{N} \sum_{i=1}^{n-1} \phi(x_i(t))$$ (6.19)

For simplicity, let us consider the zeroth moment, $m_0(t)$, ($i.e.$, $\phi(x) = x^0 = 1$), given by:

$$m_0(t) = \int_0^\infty c(x,t)dx \sim \frac{n - 1}{N}$$ (6.20)

Clearly we want this quantity (and other predicted functionals) to remain the same when we alter the particle system. We maintain a constant number of particles by picking one particle randomly to be duplicated to restore the number of particles to $n$. To ensure that our new predicted zeroth moment is still equal to $\frac{n - 1}{N}$ we have to update our normalisation parameter, $N$. Our updated normalisation parameter, $N'$, is given according to the equality:

$$\frac{n - 1}{N} = \frac{n}{N'}$$ (6.21)
i.e., we have:

\[ N' = N \frac{n}{n-1}. \]  

(6.22)

We have now ensured that the predicted zeroth moment of the system remains constant when we perform our system update. What about the other functionals of interest? In fact, this same adjustment of \( N \) ensures that the expectation of any other functional is constant. The value of the functional \( \Phi(t) \) before the update (but after the coagulation step) is given by:

\[ \Phi(t) \sim \frac{1}{N} \sum_{i=1}^{n-1} \phi(x_i(t)). \]  

(6.23)

The value after the update is given by:

\[ \Phi(t) \sim \frac{1}{N'} \left( \phi(x_j(t)) + \sum_{i=1}^{n-1} \phi(x_i(t)) \right), \]  

(6.24)

where \( j \) \((1 \leq j \leq n - 1)\) is the index of the particle that has been chosen to fill the gap. On equating the expectations of these two quantities we find:

\[
\begin{align*}
E \left[ \frac{1}{N} \sum_{i=1}^{n-1} \phi(x_i(t)) \right] &= E \left[ \frac{1}{N'} \left( \phi(x_j(t)) + \sum_{i=1}^{n-1} \phi(x_i(t)) \right) \right] \\
\frac{1}{n-1} \sum_{j=1}^{n-1} \left[ \frac{1}{N} \sum_{i=1}^{n-1} \phi(x_i(t)) \right] &= \frac{1}{n-1} \sum_{j=1}^{n-1} \left[ \frac{1}{N'} \left( \phi(x_j(t)) + \sum_{i=1}^{n-1} \phi(x_i(t)) \right) \right] \\
\frac{1}{N(n-1)} \left( n-1 \sum_{i=1}^{n-1} \phi(x_i(t)) \right) &= \frac{1}{N'(n-1)} \times \\
& \quad \left[ \sum_{j=1}^{n-1} \phi(x_j(t)) + (n-1) \sum_{i=1}^{n-1} \phi(x_i(t)) \right] \\
& = \frac{1}{N'(n-1)} \left[ \frac{n-1}{n} \sum_{i=1}^{n-1} \phi(x_i(t)) \right] \\
& = \frac{1}{N} = \frac{n}{N'(n-1)}. \end{align*}
\]

(6.25)

So we have for our updated normalisation parameter \( N' = N \frac{n}{n-1} \) as before. A similar argument can be made when, due to a stochastic jump, the particle number increases rather than decreases by one. In this case, we select uniformly
one of the \( n + 1 \) stochastic particles in the system to be removed and update the normalisation parameter to \( N' = N \frac{n}{n+1} \).

### 6.6 Test Case

Consider the case where coalescence and breakage occur according to:

\[
\begin{align*}
K(x, y) &= 1, \quad (6.26) \\
f(x, y) &= \frac{2}{x + y}, \quad (6.27) \\
i.e., \quad g(x) &= 1, \quad (6.28) \\
\beta(x, y) &= \frac{1}{x}. \quad (6.29)
\end{align*}
\]

The population balance then takes the form:

\[
\begin{align*}
\frac{\partial c(x, t)}{\partial t} &= -c(x, t) + 2 \int_{x}^{\infty} c(y, t) \frac{1}{y} dy \\
&\quad - c(x, t) \int_{0}^{\infty} c(y, t) dy + \frac{1}{2} \int_{0}^{x} c(x - y, t) c(y, t) dy. \quad (6.30)
\end{align*}
\]

We use the initial condition:

\[
c(x, 0) = \begin{cases} 
1 & : \ x = 1 \\
0 & : \text{otherwise,} 
\end{cases} \quad (6.31)
\]

and use the method of moments to find (see Appendix A):

\[
\begin{align*}
m_0(t) &= \frac{2}{1 + e^{-t}} \\
m_1(t) &= 1 \\
m_2(t) &= 3 - 2e^{-t/3} \\
m_3(t) &= 18 - 36e^{-t/3} + 19e^{-t/2} \\
m_4(t) &= 165 - 69e^{-3t/5} + 760e^{-t/2} - 675e^{-t/3} - 180e^{-2t/3} \\
m_5(t) &= \frac{4095}{2} - \frac{21853}{2} e^{-2t/3} - 5175e^{-3t/5} + 26220e^{-t/2} \\
&\quad - 14445e^{-t/3} - 180te^{-2t/3} + 2280e^{-5t/6}. \quad (6.32)
\end{align*}
\]
This is a convenient analytical solution to use for testing the stochastic algorithm. Note that we can treat all quantities as dimensionless in this case.

In this test case we examine the moments as they can describe several quantities of interest when simulating liquid-liquid extraction problems. The zeroth moment, \( m_0(t) \), indicates the total number of droplets in the system and the first moment, \( m_1(t) \), indicates the total holdup (unchanging in this mass-preserving situation). Another quantity of interest is the Sauter mean diameter, \( d_{32} \), which indicates the total interfacial area available for mass transfer, and is given by the ratio of the third moment to the second moment on a diameter basis:

\[
d_{32} = \frac{\int_0^\infty d^3 c(d) dd}{\int_0^\infty d^2 c(d) dd} = \left( \frac{6}{\pi^2} \right)^{1/3} \frac{m_1(t)}{m_2(t)}
\]

Thus, \( d_{32} \) is given by the first moment and the “two-thirds” moment when taken on a volume basis. The latter of these quantities is easy to predict with either of the stochastic algorithms under investigation, but is not straightforward to calculate for the test case. It is however a “less than one” moment, thus sharing important characteristics with the zeroth moment, \( m_0(t) \), which will be of importance when being simulated with the Mass Flow Algorithm.

### 6.7 Numerical Results

Figure 6.6 and Figure 6.7 show sample results of stochastic simulation of the coalescence-breakage equation for the test case described above. The results are averaged over a number of repeated runs, \( L \), and confidence bounds are derived (e.g. Goodson and Kraft, 2002) within which we can be 99.9% sure that the correct solution lies.

It is immediately evident that DSA and MFA have varying success when it comes to predicting different moments of the particle size distribution. For the zeroth moment, despite a much larger particle number (\( 2^{17} \) as opposed to \( 2^{11} \)), MFA has very wide confidence intervals that do not always encompass the analytical solution. DSA on the other hand, even for this moderate particle number, gives a good prediction of the zeroth moment. For the fifth moment, the situation is reversed. Now (with the same particle number, \( 2^{11} \)) MFA gives a much better prediction of the fifth moment, while the confidence bands for DSA
are large and fluctuate widely.

This behaviour can be explained by considering the way the functionals of the particle size distribution are approximated (recall Equations 3.22 and 6.10):

\[
\int_0^\infty \phi(x)c(x)dx \sim \frac{1}{N} \sum_{i=1}^{n} \phi(x_i) \quad \text{or} \quad \frac{1}{N} \sum_{i=1}^{n} \frac{\phi(x_i)}{x_i}.
\]

When approximating the zeroth moment using MFA, it is actually the sum of the \(-1\)th powers of the stochastic particle sizes that is calculated and therefore the smallest particles have most influence over the result. The appearance, due to breakage, or disappearance, due to coalescence, of a very small particle will result in a large step change in the zeroth moment. To some extent, MFA is self compensating in this regard. The coalescence step is based on a rate \(K(x_i,x_j)\); the \(\frac{1}{x_j}\) factor means that small particles are more likely to be chosen for coalescence than they are in DSA. Also, in choosing a new particle size in a breakage step, the \(y\beta(x_i,y)\) distribution (as opposed to \(\beta(x_i,y)\)) means that a small particle is less likely to be produced than it is in DSA. But even with these compensatory mechanisms, the varying presence of very small particles causes large fluctuations in the prediction of the zeroth moment. This is a problem that was not encountered when the mass flow algorithm was studied for the coagulation-only case (Eibeck and Wagner, 2001). Without breakage, there is no opportunity for the production of very small particles, so no large fluctuations in zeroth moment are observed.

The Sauter mean diameter, as previously mentioned in Section 6.6, relies on the simulation of the “two-thirds” moment, \(m_{\frac{2}{3}}(t)\), which, like the zeroth moment, will suffer the same fluctuations when predicted by MFA. In this case, it is the sum of the \(-1/3\) powers of the stochastic particle sizes that predicts \(m_{\frac{2}{3}}(t)\), so the appearance and disappearance of small particles will again lead to fluctuations in the prediction of \(d_{\frac{3}{2}}\).

The higher moments are where the successful variance reduction of the mass flow algorithm begins to show. For the same particle number—and similar CPU times—MFA exhibits much narrower confidence bands than for DSA. Thus, based on these initial observations, it would seem that DSA holds the advantage for the lower (<1) moments and MFA is more efficient at predicting higher (≥1)
Figure 6.6: Comparison of the zeroth moment predicted by Direct Simulation (DSA) and Mass Flow (MFA).

6.7.1 Convergence Properties

It is useful to consider the convergence properties of the two algorithms, i.e., how does the systematic error of the predicted functionals depend on the particle number, \( n \)? This will then give an idea of the CPU time required for each method to produce a fixed error and will indicate which is faster. For these studies, we approximate the systematic error by the average error over the simulation time, and the statistical error by the maximum width of the confidence interval within that time (Goodson and Kraft, 2002).

For the MFA case, Figure 6.8 shows the convergence properties of the 2nd to 5th moments. As expected, the systematic error decreases as the inverse of the particle number. In order to produce this figure, it is necessary to perform simulations where the statistical error (denoted by the error bars) is small enough that the trend in systematic error can be detected. This, for the MFA case,
Figure 6.7: Comparison of the fifth moment predicted by Direct Simulation (DSA) and Mass Flow (MFA).

involves a very small number of particles and a very large number of trajectories generated.

A somewhat different result occurs when we examine the convergence properties of the zeroth moment predicted by the mass flow algorithm (Figure 6.9). Using the same product of number of particles and number of runs (104857600), the error bars are now so small as not to show up on the plot for low particle numbers. Furthermore, the convergence is much slower; the systematic error is decreasing approximately as $n^{-1/3}$.

An illustration of what this convergence behaviour actually means is shown in Figure 6.10 for particle numbers of $2^5 \rightarrow 2^8$. Here, the large number of repeated runs removes the fluctuations in the predicted zeroth moment, thus showing the large systematic error in the results.

When we examine the same convergence properties for the direct simulation algorithm, we find that all the moments converge at the same rate, proportional to $\frac{1}{n}$ (Figure 6.11). From this study we can now make a comparison of the two algorithms: which algorithm takes less CPU time to predict a given moment with
Figure 6.8: Convergence to the analytical solution of the 2nd to 5th moments predicted by the mass flow algorithm. The solid lines show slopes of -1. In this study, $n \times L$ is kept constant at 104857600.
Figure 6.9: Convergence to the analytical solution of the zeroth moment predicted by the mass flow algorithm. The solid line shows a slope of \(-1/3\). In this study, \(n \times L\) is kept constant at 104857600.
Figure 6.10: Convergence to the analytical solution of the zeroth moment predicted by the mass flow algorithm. In this study, \( n \times L \) is kept constant at 104857600. The number of repeated runs is sufficient to smooth out any fluctuations in the confidence intervals, but the systematic error is still significant.
a given error?

6.7.2 DSA or MFA?

Figures 6.12 – 6.14 give further evidence to that suggested by the initial inspection of the simulation results: MFA gives better predictions for higher moments, but DSA is better for predicting the zeroth moment. The difference in CPU time is up to two orders of magnitude in favour of MFA for the higher moments, and is considerably more than that in favour of DSA for the zeroth moment.

6.8 Minimum Particle Size

Frequently, a real system in which coalescence and breakage occur will have a minimum particle size. This can be a well-defined monomer unit as in the case of polymerisation-depolymerisation (e.g. Ziff and McGrady, 1985), or can be
Figure 6.12: Single run CPU time required to predict the zeroth moment with a given systematic error.
Figure 6.13: Single run CPU time required to predict the second moment with a given systematic error.
Figure 6.14: Single run CPU time required to predict the fifth moment with a given systematic error.
introduced, due to stability considerations, as the size below which any particles quickly coalesce to produce a larger, stable, particle (e.g. Liu and Li, 1999). As well as better approximating a real system, incorporating this minimum particle size into the simulation algorithm will have a positive effect on the viability of using the mass flow algorithm to predict the zeroth moment; having fewer very small particles will reduce the size of the fluctuations in the zeroth moment. To examine this effect, we impose a minimum stable particle size on the test case and simulate the zeroth moment using both DSA and MFA. There is no longer a simple analytical solution for the moments, so we instead compare the results predicted by MFA with the converged solution predicted by DSA (which has been shown to be reliable).

There are two ways of considering how the minimum particle size, \( x_{\text{min}} \), affects the simulation:

1. No particle smaller than \( x_{\text{min}} \) can exist. Thus we can infer that breakage of a particle smaller than \( 2x_{\text{min}} \) cannot occur. This would be the case if there were a well-defined monomer unit (but note also that if each particle is made up of an integer number of monomer units the problem would be discrete rather than continuous).

2. A particle smaller than \( x_{\text{min}} \) can exist and can take part in coalescence events, but will not break into smaller particles. This is the approach used by e.g. Tsouris and Tavlarides (1994); they refer to this minimum size as a maximum stable drop diameter, i.e., the largest drop size for which no breakage can occur.

### 6.8.1 Case 1

For case 1 we need to alter the probability density function used to pick the sizes of daughter particles when a breakage event occurs. We no longer allow any particle smaller than \( x_{\text{min}} \) or larger than \( x - x_{\text{min}} \) to be formed by breakage of a particle of size \( x \). Clearly, what was previously a uniform distribution over the size range \([0, x]\) becomes a uniform distribution over the range \([x_{\text{min}}, x - x_{\text{min}}]\).
so we have:

\[
\beta(x, y) = \begin{cases} 
\frac{1}{x - 2x_{\min}} & : x_{\min} < y < x - x_{\min} \\
0 & : \text{otherwise.}
\end{cases} \quad (6.35)
\]

This leads to, for MFA simulation:

\[
\frac{y\beta(x, y)}{\int y\beta(x, y)\,dy} = \begin{cases} 
\frac{2y}{x(x - 2x_{\min})} & : x_{\min} < y < x - x_{\min} \\
0 & : \text{otherwise.}
\end{cases} \quad (6.36)
\]

In both these cases, setting \(x_{\min} = 0\) returns us to the case where no minimum size is imposed.

Also, as mentioned above, we must only choose particles larger than \(2x_{\min}\) for breakage, so as not to produce particles smaller than \(x_{\min}\).

6.8.2 Case 2

For case 2, it is not necessary to alter the daughter particle size distribution. Our only restriction in this case is not to pick any particle smaller than \(x_{\min}\) for breakage. This is the most common way a minimum particle size is imposed for liquid-liquid systems, so this is the case we examine. The only alteration that needs to be made to the simulation algorithm is to add a null event in the case that a particle smaller than \(x_{\min}\) is chosen for breakage.

Figure 6.15 shows a comparison of the zeroth moment predicted by MFA with the converged solution predicted by DSA for various minimum sizes. In this study, MFA is performed with particle number \(n = 2^{15}\) and only 50 repetitions. The large systematic error has now decreased to the point where the converged solution lies within the confidence bounds, even though the fluctuations are quite large. As the minimum particle size increases, the fluctuations decrease, and for \(x_{\min} = 1\), MFA gives a good prediction of the zeroth moment. This is consistent with the results of Eibeck and Wagner (2001) for the coagulation only case, where no particles smaller than \(x = 1\) occur. The single run simulation CPU time, \(t_{sr}\), is now of the order of a minute.
Figure 6.15: Behaviour of the Mass Flow Algorithm with the imposition of a minimum particle size, $x_{\text{min}}$. The confidence bounds are for the solutions given by MFA. The solid lines are the converged solutions given by DSA.
6.9 Ternary Breakage

Although breakage in most physical systems is assumed to be binary, the problem of ternary (and higher) breakage has been observed experimentally (e.g. Modes, 1999; Simon, 2004, in liquid-liquid systems) and modelled, mainly from a statistical point of view. Davis (1989) considered two approaches: the cascade model, where a succession of binary breakages occur, with each daughter particle continuing to break; and the remaining piece breakage model, where on each binary breakage step, one piece is kept and the other breaks further. Hill and Ng (1996) considered the statistical requirements of any classes of functions used to describe multiple breakage. They derived several sets of daughter distribution functions with the property they called exchangeability, i.e. the distribution function should not depend on which particle volume is first, second or third.

In the work of Modes (1999); Simon (2004), an average (non-integer) number of daughter droplets was assumed to be formed on breakage. However, with our stochastic simulation we have the ability to simulate each breakage event individually, modelling some as binary breakage and some as ternary (or higher).

The remaining piece breakage model of Davis (1989) assumes that each successive binary breakage step obeys the same (self-similar) daughter distribution function. It is therefore clear that the daughter particles so produced would not satisfy the exchangeability requirement of Hill and Ng (1996). However, the remaining piece breakage procedure is one that is amenable to stochastic simulation: we just need to find a suitable sequence of daughter distribution functions to ensure that exchangeability is satisfied.

We use the following notation for the daughter droplet size distribution. If a droplet of size $x$ undergoes breakage to form $\nu$ daughter droplets, then the probability that a daughter particle is formed in the size range $[y \rightarrow y + dy]$ is given by the probability density function $\beta_\nu(x, y)dy$.

Recall that for binary breakage of a particle of size $x$, once we have selected the first daughter particle to be size $y_1$, then the second daughter particle is automatically chosen (with unit probability) to be of size $y_2 = x - y_1$. We require that the distribution of $y_1$ be the same as that of $y_2$, but it can be easily shown that the only restriction this places on the daughter size distribution function is
that it be symmetric on the range \([0, x]\):
\[
\beta_2(x, y_2) = \beta_2(x, y_1) = \beta_2(x, x - y_2),
\]
(6.37)
thus we have \(\beta_2(x, y_2) = \beta_2(x, x - y_2)\) and symmetry is all we require. This is the
condition of exchangeability for binary breakage.

With ternary breakage, daughter size selection is not so straightforward. When the first
daughter particle has been picked, there are still two particles that need to be selected
on the size range \([0, x - y_1]\), so we need a size distribution for the second daughter
particle. We denote this size distribution by \(\beta_2(x - y_1, y_2)\). An argument similar
to the one above shows that \(\beta_2(x - y_1, y_2)\) must be symmetric on the range
\([0, x - y_1]\). We also require that the second and third daughter particles are selected
according to the same distribution as the first, i.e.:
\[
\beta_3(x, y_2) = \int_0^{x-y_2} \beta_2(x - y_1, y_2) \beta_3(x, y_1) dy_1
\]
(6.38)
We now take the form of the Beta distribution used by Bahmanyar and Slater (1991),
which depends on the number of daughter droplets formed, \(\nu\):
\[
\beta_\nu(x, y_1) = (\nu - 1) \left[ 1 - \left( \frac{y_1}{x} \right) \right]^{(\nu-2)} \frac{1}{x}.
\]
(6.39)
Note that we have modified their volume distribution to give a number distribution. It can
easily be seen that setting \(\nu = 2\) gives the familiar uniform daughter density
distribution for binary breakage. For ternary breakage, \(\nu = 3\), we find
the following:
\[
\beta_3(x, y_1) = \frac{2}{x} \left( 1 - \frac{y_1}{x} \right),
\]
(6.40)
so we can use equation 6.38 to find the size distribution for the second daughter
droplet:
\[
\beta_3(x, y_2) = \int_0^{x-y_2} \beta_2(x - y_1, y_2) \frac{2}{x} \left( 1 - \frac{y_1}{x} \right) dy_1 = \frac{2}{x} \left( 1 - \frac{y_2}{x} \right).
\]
(6.41)
By using \(\beta_2(x - y_1, y_2) = \frac{1}{x - y_1}\) we can satisfy the above criterion, and also the
criteria that \(\beta_2\) is symmetric on the range \([0, x - y_1]\) and that it must be a
probability density function, \( i.e. \):
\[
\int_0^{x-y} \beta_2(x - y_1, y_2)dy_2 = 1. \tag{6.42}
\]

Note that the probability density function for choosing the second daughter particle has exactly the same form as that given for binary breakage by Bahman-yar and Slater’s Beta distribution. In fact, it is true for any \( \nu > 2 \) that:
\[
\int_0^{x-y} \frac{\nu - 2}{x - y'} \left( 1 - \frac{y}{x - y'} \right)^{\nu-3} \frac{\nu - 1}{x} \left( 1 - \frac{y'}{x} \right)^{\nu-2} dy' = \frac{\nu - 1}{x} \left( 1 - \frac{y}{x} \right)^{\nu-2} \tag{6.43}
\]
i.e.:
\[
\beta_{\nu}(x, y) = \int_0^{x-y} \beta_{\nu-1}(x - y', y)\beta_{\nu}(x, y')dy' \quad \nu = 2, 3, \ldots, \tag{6.44}
\]
with \( \beta_1(x, y) = \delta(y - x) \).

In fact, the Beta distribution as stated in Equation (6.39) is a specific example of a more general class of functions, derived by Hill and Ng (1996) and shown by Diemer and Olson (2002c) to relate to the general form of the Beta distribution, also used by Cabassud et al. (1990) to model droplet breakage:
\[
\beta_p(x, y) = \frac{(\nu(p + 1) - 1)!}{p!(\nu - 1)(p + 1) - 1)!} x^{p+1} \left( 1 - \frac{y}{x} \right)^{(\nu-1)(p+1)-1}. \tag{6.45}
\]
Here \( p \) is a parameter. Setting \( p = 0 \) reduces to the Beta distribution above, and letting \( p \to \infty \) gives a series of delta functions (Diemer and Olson, 2002c):
\[
\beta_{\nu}^{\infty}(x, y) = \delta(y - x/\nu) \tag{6.46}
\]
See Appendix B for proof that this class of functions satisfies the required constraint (Equation (6.44)).

Figure 6.16 shows the daughter droplet distributions for binary breakage for different values of the parameter \( p \) and for \( \nu \)-nary breakage for the case \( p = 0 \). For binary breakage, it can be seen that with increasing \( p \), the distribution becomes narrower, centred around a daughter volume of \( y = x/2 \).

We can also write the general form of the daughter distribution function using
Figure 6.16: Predicted daughter droplet size distributions according to Equation (6.45).
a size fraction, \(0 < f_y < 1\), where \(f_y = y/x\). We use the fact that:

\[
\beta^p_v(x, y) dv = \beta^v_y(df_y),
\]

(6.47)

to give:

\[
\beta^v_y(df_y) = \frac{\nu(p + 1) - 1)!}{p!((\nu - 1)(p + 1) - 1)!} f_y^p (1 - f_y)^{(\nu-1)(p+1)-1}
\]

(6.48)

We can now modify our breakage algorithm for ternary breakage. Once a droplet of size \(x\) has been selected for breakage, we choose the first daughter droplet \(y_1 = f_y_1 x\) according to the distribution \(\beta_3(f_y_1)\). We then choose the second daughter droplet of size \(y_2 = f_y_2(x - y_1)\) according to the distribution \(\beta_2(f_y_2)\). The third daughter droplet is then (with probability one) of size \(y_3 = x - y_1 - y_2\). For a constant-\(n\) Monte Carlo simulation we will need to remove two particles from the stochastic particle array and replace them with the second and third daughter droplets and also modify the particle number \(N\) accordingly.

For simulation purposes, we may wish to use the inversion method to generate the sizes of daughter particles. For this we need to integrate the general form of the daughter distribution function:

\[
\int_0^{f_y} \beta^p_v(f'_y) df'_y = \frac{B(f_y; p + 1, (\nu - 1)(p + 1))}{B(p + 1, (\nu - 1)(p + 1))},
\]

(6.49)

where \(B(x; m, n)\) is the incomplete Beta function (e.g. Whittaker and Watson, 1990):

\[
B(x; m, n) = \int_0^x y^{m-1}(1 - y)^{n-1} dy,
\]

(6.50)

and \(B(m, n)\) is the Beta function:

\[
B(m, n) = \int_0^1 y^{m-1}(1 - y)^{n-1} dy = \frac{\Gamma(m)\Gamma(n)}{\Gamma(n + m)} = \frac{(m - 1)!(n - 1)!}{(n + m - 1)!}.
\]

(6.51)

Recall that the random deviate (in this case \(f_y\)) is chosen such that:

\[
r = \int_0^{f_y} \beta^p_v(f'_y) df'_y,
\]

(6.52)

where \(r\) is a uniform random variable on \([0, 1]\). For the case \(p = 0\), equation 6.52 can be simplified and inverted such that a daughter particle size can be generated
according to:

\[ f_y = 1 - r^{1/(\nu-1)}. \]  

(6.53)

In the general case for \( p \neq 0 \), the form of equation 6.50 suggests that a table of values for the incomplete Beta function (e.g. Pearson, 1968) may be needed to choose the daughter particle sizes.

One question that remains is how well do these theoretical distributions predict reality? Surface energy arguments predict that a large and small droplet are more likely to be formed on breakage than two equi-sized droplets. Hill and Ng (1996) could not find a single daughter distribution function that satisfied this requirement, but solved the problem by considering linear combinations of functions such as Equation (6.45) with varying parameters. See also Wang et al. (2003) for further discussions of the merits of phenomenological models versus assumed functions. A further twist on the above statistical arguments is provided by Diemer and Olson (2002c), who contend that the criterion of exchangeability need not be satisfied in some physical cases. Their example of a droplet undergoing multiple breakage by elongation into a rod-like structure that breaks into two major daughter droplets with a number of smaller satellite droplets suggests that not all products of particle breakage are necessarily statistically identical. They propose representing multiple breakage as a rapid sequence of binary breakage events, each of which may or may not have the exchangeability property.

A Solution of Test Case

To solve the equation:

\[
\frac{\partial c(x, t)}{\partial t} = -c(x, t) + 2 \int_x^\infty c(y, t) \frac{1}{y} dy - c(x, t) \int_0^\infty c(y, t) dy + \frac{1}{2} \int_0^x c(x - y, t) c(y, t) dy, \tag{6.54} \]

}\]
we use the method of moments (e.g. Frenklach, 1985), i.e., we multiply by $x^k$ and integrate over all $x$:

$$
\int_0^\infty x^k \frac{\partial c(x,t)}{\partial t} \, dx = \frac{d}{dt} m_k(t) = - \int_0^\infty x^k c(x,t) \, dx + 2 \int_0^\infty x^k \int_x^\infty c(y,t) \frac{1}{y} \, dy \, dx
- \int_0^\infty x^k c(x,t) \int_0^\infty c(y,t) \, dy \, dx + \frac{1}{2} \int_0^\infty x^k \int_0^x c(x-y,t) c(y,t) \, dy \, dx.
$$

(6.55)

Treating each term on the RHS in turn:

$$
\int_0^\infty x^k c(x,t) \, dx = m_k(t). 
$$

(6.56)

$$
\int_{x=0}^\infty x^k \int_{y=x}^\infty c(y,t) \frac{1}{y} \, dy \, dx = \int_{y=0}^\infty c(y,t) \frac{1}{y} \int_{x=0}^y x^k \, dx \, dy
= \int_{y=0}^\infty c(y,t) \frac{1}{y} \frac{y^{k+1}}{k+1} \, dy
= \int_{y=0}^\infty c(y,t) \frac{y^k}{k+1} \, dy = \frac{m_k(t)}{k+1}.
$$

(6.57)

$$
\int_{x=0}^\infty x^k c(x,t) \int_{y=0}^\infty c(y,t) \, dy \, dx = m_k(t)m_0(t).
$$

(6.58)

$$
\int_{x=0}^\infty x^k \int_{y=0}^x c(x-y,t)c(y,t) \, dy \, dx
= \int_{y=0}^\infty c(y,t) \int_{x=y}^\infty x^k c(x-y,t) \, dy \, dx
= \int_{y=0}^\infty c(y,t) \int_{z=0}^\infty (z+y)^k c(z,t) \, dy \, dz
= \int_{y=0}^\infty c(y,t) \int_{z=0}^\infty \sum_{r=0}^k \binom{k}{r} z^{k-r} y^r c(z,t) \, dy \, dz
= \sum_{r=0}^k \binom{k}{r} \int_{y=0}^\infty c(y,t) y^r \, dy \int_{z=0}^\infty c(z,t) z^{k-r} \, dz
= \sum_{r=0}^k \binom{k}{r} m_r(t)m_{k-r}(t).
$$

(6.59)
We then have differential equations for $m_k(t)$:

\[
\frac{d m_0(t)}{d t} = m_0(t) - \frac{m_0^2(t)}{2} \quad (6.60)
\]
\[
\frac{d m_1(t)}{d t} = 0 \quad (6.61)
\]
\[
\frac{d m_k(t)}{d t} = \frac{1 - k}{1 + k} m_k(t) + \frac{1}{2} \sum_{r=1}^{k-1} \binom{k}{r} m_r(t) m_{k-r}(t) \quad k = 2, 3, \ldots \quad (6.62)
\]

For the initial condition:

\[
c(x, 0) = \begin{cases} 
1 : x = 1 \\
0 : \text{otherwise},
\end{cases} \quad (6.63)
\]

i.e., $m_k(0) = 1, \quad k = 0, 1, 2, \ldots$, we have a series of first order differential equations that are separable (6.60), trivial (6.61) or have constant coefficients once the lower order moments have been found (6.62). These can be solved to give:

\[
m_0(t) = \frac{2}{1 + e^{-t}}
\]
\[
m_1(t) = 1
\]
\[
m_2(t) = 3 - 2e^{-t/3}
\]
\[
m_3(t) = 18 - 36e^{-t/3} + 19e^{-t/2}
\]
\[
m_4(t) = 165 - 69e^{-3t/5} + 760e^{-t/2} - 675e^{-t/3} - 180e^{-2t/3}
\]
\[
m_5(t) = \frac{4095}{2} - \frac{21853}{2} e^{-2t/3} - 5175e^{-3t/5} + 26220e^{-t/2} - 14445e^{-t/3} - 180e^{-2t/3} + 2280e^{-5t/6} \quad (6.64)
\]

B  Algebra relating to ternary breakage

We wish to show that:

\[
\int_0^{x-y} \frac{\nu - 2}{x-y'} \left(1 - \frac{y}{x-y'}\right)^{\nu-3} \frac{\nu - 1}{x} \left(1 - \frac{y'}{x}\right)^{\nu-2} dy' = \frac{\nu-1}{x} \left(1 - \frac{y}{x}\right)^{\nu-2}.
\quad (6.65)
\]
CHAPTER 6. COALESCENCE AND BREAKAGE

Noting that:
\[
\left(1 - \frac{y}{x - y'} \right) \left(1 - \frac{y'}{x} \right) = \left(\frac{x - y' - y}{x - y'} \right) \left(\frac{x - y'}{x} \right) = \frac{x - y' - y}{x},
\]
the integral becomes:
\[
\int_0^{x-y} \frac{(\nu - 1)(\nu - 2)}{x(x-y')} \left(\frac{x - y' - y}{x}\right)^{\nu-3} \left(\frac{1}{x-y'}\right) dy' = x^{1-\nu} (\nu - 1)(\nu - 2) \int_0^{x-y} (x - y - y')^{\nu-3} dy'
\]
\[
= x^{1-\nu} (\nu - 1)(\nu - 2) \nu - 2 = \frac{\nu - 1}{x} \left(1 - \frac{y'}{x}\right)^{\nu - 2}
\]
as required.

For the more general case we want Equation (6.44) to be true for the function in Equation (6.45), i.e.:
\[
\frac{(\nu(p + 1) - 1)!}{p!(\nu - 1)(p + 1) - 1)!} \frac{y^p}{x^{p+1}} \left(1 - \frac{y}{x}\right)^{(\nu-1)(p+1)-1}
\]
\[
\int_0^{x-y} \frac{(\nu - 1)(p + 1) - 1)!}{p!(\nu - 2)(p + 1) - 1)!} \frac{y^p}{(x - y')^{p+1}} \left(1 - \frac{y}{x-y'}\right)^{(\nu-2)(p+1)-1} \times
\]
\[
\frac{(\nu(p + 1) - 1)!}{p!(\nu - 1)(p + 1) - 1)!} \frac{y^p}{x^{p+1}} \left(1 - \frac{y'}{x}\right)^{(\nu-1)(p+1)-1} dy'
\]
Cancelling terms gives:
\[
\left(1 - \frac{y}{x}\right)^{(\nu-1)(p+1)-1} = \frac{(\nu - 1)(p + 1) - 1)!}{p!(\nu - 2)(p + 1) - 1)!} \times
\]
\[
\int_0^{x-y} \frac{y^p}{(x - y')^{p+1}} \left(1 - \frac{y}{x-y'}\right)^{(\nu-2)(p+1)-1} \left(1 - \frac{y'}{x}\right)^{(\nu-1)(p+1)-1} dy'
\]
which reduces to:
\[
\left(1 - \frac{y}{x}\right)^{(\nu-1)(p+1)-1} = \frac{(\nu - 1)(p + 1) - 1)!}{p!(\nu - 2)(p + 1) - 1)!} \times
\]
\[
\int_0^{x-y} \frac{y^p}{x^{p+1}} \left(\left(\frac{x - y' - y}{x - y'}\right) \left(\frac{x - y'}{x}\right)\right)^{(\nu-2)(p+1)-1} dy'
\]
The integral can now be written in a familiar form:

\[
\left(1 - \frac{y}{x}\right)^{(\nu - 1)(p+1) - 1} = \frac{((\nu - 1)(p + 1) - 1)!}{p!(\nu - 2)(p + 1) - 1)!} \left(1 - \frac{y}{x}\right)^{(\nu - 1)(p+1) - 1} \times
\int_0^1 \left(\frac{y'}{x - y}\right)^p \left(1 - \frac{y'}{x - y}\right)^{(\nu - 2)(p+1) - 1} d \left(\frac{y'}{x - y}\right) \tag{6.71}
\]

The integral takes the value \(\frac{p!(\nu - 2)(p+1) - 1)!}{((\nu - 1)(p+1) - 1)!}\), which is the inverse of the first fraction on the right hand side, thus the equivalence is shown.
Chapter 7

Simulation of a 2-Component RDC

7.1 Modelling Coalescence and Breakage

Mathematical modelling of liquid-liquid extraction can be categorised into three basic types. Empirical fitting is used, based on experimental results, to predict mean drop size or hold-up based on liquid properties, column geometry and operating conditions (e.g. Kumar and Hartland, 1986, for a rotating disc contactor). In Stagewise modelling, an extraction column is described as a series of perfectly mixed stages; effectively, each stage is treated as a CSTR. These stages may be real stages, e.g. as in a sieve-plate column, or cascades to approximate a differentially varying system (e.g. a packed extraction column). Differential models involve formulation of differential conservation equations for the two liquid phases. Within this category, there are two further distinct mathematical treatments of the liquid phases. The dispersed phase can be treated as pseudo-homogeneous, thus in effect there are two continuous phases in the model. When coalescence and breakage cannot be neglected, and the dispersed phase continually undergoes changes, the pseudo-homogeneous treatment breaks down, and a population balance model must be applied. Here a differential balance is formulated for the number of drops in the dispersed phase, taking into account birth and death rates due to such phenomena as coalescence, breakage and convective transport.

Further description of the models mentioned above, as well as information on
many types of liquid-liquid extraction equipment, can be found in the extensive review by Mohanty (2000).

7.2 Coalescence and Breakage Models

Here, we briefly review the available models for coalescence and breakage rates found in the literature. For clarity, we present the models based on the functional dependence on the droplet size (either diameter, \(d\), or volume, \(v\)). We have not explicitly written out the dependence on other important parameters, such as liquid properties, column geometry or operating conditions. For the full expressions, we refer back to the literature from which we have taken the models.

7.2.1 Turbulent Dispersions

Some of the simplest models for coalescence and breakage rates are empirical. Simple functions (normally power law relationships) are proposed for the dependence of coalescence or breakage on size, and experiments are performed to determine the various unknown parameters. Some proposed relationships may have a degree of physical reasoning, such as being proportional to diameter, surface area or volume (Valentas et al., 1966), while others have no physical basis and are purely empirical (e.g. Balmelli et al., 2000). While this method has the advantage of being very computationally efficient, a different set of experiments must be performed for each different set of operating conditions; values found for any empirical parameters are only valid for a limited set of applications.

If a theoretical formulation of the relevant kinetic rates is preferred, then the works of Coulaloglou and Tavlarides (1976, 1977) are frequently considered as a reference point. The original formulation for the coalescence rate was:

\[
K(d_1, d_2) = c_1(d_1^2 + d_2^2)(d_1^{2/3} + d_2^{2/3})^{1/2} \exp \left[ -c_2 \left( \frac{d_1 d_2}{d_1 + d_2} \right)^4 \right] \quad (7.1)
\]

The model, of a collision rate (the first two functional dependencies on \(d_1, d_2\)) multiplied by a coalescence efficiency (the exponential term) borrows from the kinetic theory of gases and is a basis that has been used (and updated) for many subsequent theoretical formulations. The first bracketed term however, refers to
a collisional cross-sectional area, which, as subsequent authors have pointed out should be \((d_1 + d_2)^2\) (corrected without comment by Hsia and Tavlarides (1983) and Bapat et al. (1983) and mentioned by e.g. Alopaeus et al. (1999)). Some other authors (e.g. Kentish et al., 1998; Maggioris et al., 2000) have used the original equations of Coulaloglou and Tavlarides in their erroneous form.

The breakage rate was formulated by consideration of the number of eddies with sufficient energy to break a droplet:

\[
g(d) = b_1d^{-2/3} \exp(-b_2d^{-5/3}) \tag{7.2}
\]

The daughter droplet size distribution function (called the breakage function in the original paper) was assumed to be normal:

\[
\beta(d_0, d) = \frac{7.2d^2}{d_0^3} \exp \left[ -4.5 \frac{(2d^3 - d_0^3)^2}{d_0^6} \right] \tag{7.3}
\]

These original equations have been updated at various points (e.g. Sovová, 1981) but the theoretical reasoning remains largely unchanged. The most recent revision of these equations was performed by Tsouris and Tavlarides (1994). They modelled the breakage rate as a collision rate between droplets and eddies (with a breakage efficiency included). By integrating over all the appropriately sized eddies, they obtained the following size dependence of the breakage rate:

\[
g(d) = b_1 \int_{2/d}^{2/d_{e, \text{min}}} (2/k + d)^2 \sqrt{8.2k^{-2/3} + 1.07d^{2/3}k^2} \exp(-b_2d^{11/3})dk \tag{7.4}
\]

(Tsouris and Tavlarides, 1994) were among the first to consider physical arguments when formulating the daughter droplet size density function. By considering surface energy requirements of the daughter droplets, they formulated the following for \(\beta(d_0, d)\) (\(d_{\text{min}}\) is the minimum size drop that can be formed on breakage):

\[
\beta(d_0, d) = \frac{d_{\text{min}}^2 + (d_0 - d_{\text{min}})^2 - d^2}{\int_0^{d_0} d_{\text{min}}^2 + (d_0 - d_{\text{min}})^2 - d^2 dd} \tag{7.5}
\]

A revision of the coalescence model gave:

\[
K(d_1, d_2) = c_1(d_1 + d_2)^2(d_1^{2/3} + d_2^{2/3})^{1/2} \exp \left[ -\frac{c_2}{(d_1 + d_2)^{2/3}} \right] \tag{7.6}
\]
A similar treatment is performed by Luo and Svendsen (1996) for the breakage rate, describing drop breakage as being due to high frequency oscillations in the neighbourhood of a bubble caused by an approaching eddy. This latter work has the advantage that it does not include any unknown parameters. The authors also make the claim that there are no assumptions made as to the probability distribution of the sizes of daughter particles formed on breakup; the daughter droplet probability density function is a result of the model rather than an input. However, the derivation of the model means that, effectively, the density function is the same as that used by Tsouris and Tavlarides, as in both cases it is a consideration of the surface energies of mother and daughter droplets that results in the final rate expression. The structure of the expression derived by Luo and Svendsen is largely the same as that of Tsouris and Tavlarides, but some differences in theoretical reasoning result in the following breakage model:

\[
g(d) = b_1 d^{-1/3} \int_{2/d_{e,\text{min}}}^{2/d_{e,\text{min}}} \left( \frac{2}{k + d} \right)^{5/3} k^{5/3} \exp(-b_2 d^2 k^{11/3}) dk \tag{7.7}
\]

Suspension polymerisation in a turbulent system was modelled by Maggioris et al. (2000) based on the approach of Coulaloglou and Tavlarides (1977). Due to more detailed consideration of the energy dissipation on fluctuation due to elasticity, they derive a somewhat complicated model for the breakage efficiency exponent, \(\lambda_g(d)\), for the inertial subrange of turbulence:

\[
\lambda_g(d) = b_2 \left[ b_3 d^{1/3} + \left( \frac{b_2^2 b_4 d^{14/3}}{2 \sqrt{1 - b_4 d^2}} \left[ 1 + \sqrt{1 - b_4 d^2} \exp \left( -\frac{4(1 - \sqrt{1 - b_4 d^2})}{b_3 b_4 d^{10/3}} \right) \right] - \frac{1}{\sqrt{1 - b_4 d^2}} \exp \left( -\frac{4(1 - \sqrt{1 - b_4 d^2})}{b_3 b_4 d^{10/3}} \right) \right]^{-1} + b_5 d^{-5/3} \right) \tag{7.8}
\]

The constants \(b_3 \rightarrow b_5\) in this expression are groups of parameters such as surface tension and liquid density. The constant \(b_2\) at the beginning of the RHS is an adjustable empirical parameter. A slightly different (but equally complicated) version of the above expression is found for the viscous dissipation range, due to a different dependence of droplet velocity on size.

The overall breakage rate (in the inertial subrange) is given as:

\[
g(d) = b_1 d^{-1/3} \exp[-\lambda_g(d)] \tag{7.9}
\]
In formulating a coalescence rate, Maggioris et al. (2000) followed the method of Coulaloglou and Tavlarides (1977), but neglected to correct the erroneous expression for the collisional cross-sectional area in the collision rate. Nevertheless, they formulated an original expression for a coalescence efficiency exponent (again in the inertial subrange):

\[ \lambda_{K}(d_1, d_2) = c_1 \lambda_g(d_{12}) - c_2 (d_1^3 + d_2^3)^{-5/3} (d_1^{-1} + d_2^{-1})^{-10/3} \]  

(7.10)

where \( d_{12} \) is the harmonic mean of \( d_1 \) and \( d_2 \):

\[ d_{12}^{-1} = d_1^{-1} + d_2^{-1} \]  

(7.11)

and \( \lambda_g(d) \) is as defined in equation 7.8.

The expressions for coalescence and breakage rates in a turbulent dispersion are often applied to liquid-liquid extraction problems in other geometries:

### 7.2.2 Oldshue-Rushton Contactor

Tsouris et al. (1994) used the breakage and coalescence rates derived by Tsouris and Tavlarides (1994) to model an Oldshue-Rushton Contactor (multistage stirred cell extraction column). The justification for this is the geometric similarity between a stirred tank contactor and an individual cell in the column being modelled.

### 7.2.3 Oscillatory Baffled Reactor

The Oscillatory Baffled Reactor (OBR) is a relatively new technology that employs oscillatory flow past annular baffles in order to enhance heat transfer, mass transfer, mixing and other unit operations. The improved control capabilities of such a device are promising both for liquid-liquid extraction operations, where knowledge of the droplet size distribution is important for predicting performance, and for operations such as suspension polymerisation where the size distribution of the final product (as opposed to an intermediate) is of interest. Ni et al. (2002) studied the breakage and coalescence characteristics of an OBR, also using the equations of Tsouris and Tavlarides (1994) as their model. They determined the four constants \( b_1, b_2, c_1, c_2 \) by fitting the model to their experimental data, and
found good results when $c_2 = 0$. Thus, they concluded that the coalescence efficiency in an OBR is unity, i.e., every collision between two droplets results in a coalescence:

$$K(d_1, d_2) = c_1(d_1 + d_2)^2(d_1^{2/3} + d_2^{2/3})^{1/2}$$  \(7.12\)

### 7.2.4 Bubble Column Reactor

While not strictly under the heading of a liquid-liquid system, many of the features of the liquid-liquid models can be applied to modelling gas-liquid systems, notably the concepts of bubble-bubble collisions for coalescence and bubble-eddy interaction for breakage, so the bubble column reactor is included in this section.

The design of Bubble Column Reactors is still largely empirical, however Shimizu et al. (2000) have proposed theoretically derived models for breakage and coalescence. Breakage is modelled as being due to bubble deformation by eddies, resulting in a breakage rate that shares several features with the model of Tsouris and Tavlarides (1994) for a turbulent dispersion:

$$g(d) = b_1 \int_{d_k/min}^{2/d_k} (2/k + d)^2 \sqrt{1.59k^{-2/3} + d^{2/3}k^2} \exp(-b_2d^{-1}k^{2/3})dk$$  \(7.13\)

Breakage is assumed to result in two equi-sized daughter droplets.

Coalescence is considered (in common with other phenomenological models) to occur as a result of two bubbles colliding, with the rate given as the product of the collision rate and the coalescence efficiency. In this case, collision is modelled as being due to three mechanisms; turbulence, buoyancy and shear. This results in the following expression for the coalescence rate:

$$K(d_1, d_2) = \left[ c_1(d_1 + d_2)^2(d_1^{2/3} + d_2^{2/3})^{1/2} + c_2(d_1 + d_2)^2\left| (c_3d_1^{-1} + d_1)^{1/2} - (c_3d_2^{-1} + d_2)^{1/2} \right| + c_4(d_1 + d_2)^3 \right] \times \exp(-c_5(d_1^{-1} + d_2^{-1})^{-5/6})$$  \(7.14\)

Here, the terms in the square brackets give collision rates for turbulence, buoyancy and shear respectively. The exponential term gives the coalescence efficiency.
7.2.5 Rotating Disc Contactor

A Rotating Disc Contactor (RDC) is a vertical column, containing an alternating series of stationary annular baffles and rotor discs. Drop breakage has been observed to occur at the level of each rotor due to the shear stress generated by the rotor. Breakage rates for the RDC case have been formulated by considering the probability of breakage at each stage of the column (Bahmanyar and Slater, 1991). These probabilities can then be used in a Monte-Carlo type simulation of drop breakage for the lean case, where coalescence can be ignored (e.g. Ghalechian, 2002). The general form of the expression found for the breakage probability, \( p \), is:

\[
\frac{p}{1-p} = b_1 \, \text{We}_m^{b_2},
\]

where the modified Weber number, \( \text{We}_m \), is given by the expression:

\[
\text{We}_m = \frac{\rho_c d (\omega^2 - \omega_{\text{crit}}^2) D_r^2}{\sigma},
\]

with \( \omega_{\text{crit}} \) being the critical rotational speed below which no breakage takes place. It should be noted that the numerical values for \( b_1 \) and \( b_2 \) are empirically derived, and hence there is some variation in the literature values. Cauwenberg et al. (1997) derive a new expression for \( \text{We}_m \):

\[
\text{We}_m = \frac{\rho_c^{0.8} \mu_c^{0.2} d D_r^{1.6} (\omega^{1.8} - \omega_{\text{crit}}^{1.8})}{\sigma},
\]

and Schmidt et al. (2003) update the values of \( b_1 \) and \( b_2 \) from experimental data. The breakage probabilities can be used in conjunction with a drop rise velocity to give a breakage rate in the column.

The daughter droplet size distribution is taken to be a beta-distribution, depending on the number of daughter drops produced (Bahmanyar and Slater, 1991). For binary breakage, this results in uniformly distributed daughter drops.

Coalescence in an RDC has also been described with an empirical rate expression. Köhler (1998) found the following expression:

\[
K(d_1, d_2) = c_1 (d_1 + d_2)^{2.33}
\]
7.2.6 Kühni Column

The Kühni Column (or rotary agitated column) is a staged column, commonly used for separation of aromatic and aliphatic hydrocarbons. Breakage in a Kühni Column has been treated in a similar way to the RDC (Section 7.2.5), i.e., by formulating a breakage probability at each stage (Gourdon et al., 1991):

\[ p(d) = \exp \left( -b_1 d^{-5/3} \right) \]  

(7.19)

The \( d^{5/3} \) dependence comes from modelling the system using another form of the Weber number, \( We = \rho \epsilon d^{2/3} d^{5/3} / \sigma \).

In an alternative work, Kentish et al. (1998) formulated a breakage rate based on the above breakage probability, noting that it contains the same exponential dependence on diameter as the breakage rate of Coulaloglou and Tavlarides (1977):

\[ g(d) = b_1 u \exp(-b_2 d^{-5/3}) \]  

(7.20)

Here, \( u \) is the velocity of a droplet of diameter \( d \). They found that the velocity was approximately a function of the inverse square of the rotor speed only, so the functional dependence on drop size can be given as:

\[ g(d) = b_1 \exp(-b_2 d^{-5/3}) \]  

(7.21)

In the same work, a coalescence rate was formulated, assuming coalescence only takes place between two drops of the same volume, \( v \):

\[ K(v) = c_1 v^{0.56} \]  

(7.22)

7.2.7 Pulsed Sieve-Plate Column

A Pulsed Sieve-Plate Column (or pulsed perforated-plate extraction column) is a staged device, where breakage probabilities can be formulated for each plate, and an identical expression to that in Section 7.2.6 is obtained: (Gourdon et al., 1991; Gourdon and Casamatta, 1991):

\[ p(d) = \exp \left( -b_1 d^{-5/3} \right) \]  

(7.23)
The constant $b_1$ takes different values depending on the type of column used. Empirical breakage rates have also been proposed:

$$g(d) = b_1 d^{b_2} \quad (7.24)$$

With the exponent $b_2$ taking values ranging from $0.1 \rightarrow 0.8$ (Garg and Pratt, 1984) to 8 (Narsimhan et al., 1979). This second, much higher, value was first used for a generic stirred dispersion, but has been applied to the sieve-plate case (Eid et al., 1991).

The daughter droplet size distribution has been variously modelled as two equi-sized drops (Garg and Pratt, 1984), three equi-sized drops (Casamatta and Vogelpohl, 1985) or a Beta-distribution with two to four daughter drops (Eid et al., 1991; Gourdon et al., 1991).

Coalescence models are rarer than breakage models, as experiments are often done at low hold-up, where coalescence is negligible. However, Gourdon and Casamatta (1991) considered the influence of mass transfer on the coalescence rate, both for continuous to discrete phase (C→D) and discrete to continuous (D→C) transfer. The coalescence rate was modelled as the product of the collision rate and the coalescence efficiency. For the C→D case, the coalescence efficiency was taken to be zero and for D→C transfer, it was taken to be unity. Thus, for the case where coalescence is active, the coalescence rate was modelled as:

$$K(d_1, d_2) = c_1(d_1 + d_2)^2(d_1 + d_2)^{1/3} \quad (7.25)$$

### 7.2.8 Packed Liquid Extraction Column

The Packed Liquid Extraction Column differs from some of the other columns described above, in that its behaviour is closer to plug-flow than to perfectly mixed. The column packing reduces axial mixing and results in improved performance. Models for this type of column are not common, but it has been studied by Hamilton and Pratt (1984). Their sectional model assumed that coalescence only takes place between droplets of the same or adjacent sizes, and hence obtained a coalescence rate as a function of only one particle size:

$$K(d) = c_1 d^{-0.36} \quad (7.26)$$
This model specific coalescence rate (similar to the model for the Kühni column in section 7.2.6) is therefore unsuitable for use in our population balance modelling.

The empirically fitted breakage rate is given as:

\[ g(d) = b_1 d^{0.37} \]  \hspace{1cm} (7.27)

### 7.2.9 Remarks

It can be seen that breakage and coalescence models for liquid-liquid systems take two forms: empirical and theoretically derived. The empirical models commonly take a power law form, with unknown constants being adjusted to fit experimental data. The theoretically derived models generally still have parameters that can be adjusted to give a good fit to experimental results. Daughter droplet size distributions are the most widely varying aspect of the models, ranging from monodisperse (two equi-sized drops always formed), normally- or beta-distributed (maximum probability of two equi-sized drops) through uniform distribution (equal probabilities) to distributions derived through surface energy considerations (greatest probability of a small and a large drop being formed). The first three of these are merely assumed shapes (often for the purpose of ease of calculation); only the fourth model has any realistic grounding to it.

The decision on which model to use depends on the likely accuracy and efficiency of simulation. Do the improved results of a more complicated model with extra adjustable parameters justify the increased computational time likely to be required? To what extent does the presence of extra adjustable parameters reduce the physicality of the model? Alopaeus et al. (2002) contended that one criterion for a good model is that it contains as few adjustable parameters as possible. They found that using the model of Sovovà (1981) for coalescence efficiency, as opposed to that of Tsouris and Tavlarides (1994), gave an improved fit but used one more adjustable parameter. They decided that the improvement was not worth the extra parameter.
7.3 Experimental Comparison

Having presented an algorithm for simulation of coalescence and breakage in liquid-liquid systems (Chapter 6) and reviewed some physical models (Section 7.2), we now wish to apply our simulation method to a real system. To illustrate the effectiveness of the Direct Simulation Algorithm, we apply it to the situation of a laboratory scale rotating disc contactor. The experimental apparatus used is as described by Modes (1999) and the system used is butylacetate (discrete phase) and water (continuous phase) (Simon, 2004).

7.4 Models

7.4.1 Rise Velocity

Correlations for droplet terminal velocities generally make use of the Morton Number, $Mo$, to describe the fluid properties:

$$ Mo = \frac{g \mu_c^4 (\rho_c - \rho_d)}{\rho_c^2 \sigma^3} = 3.0 \times 10^{-10} \text{ for butylacetate/water.} \quad (7.28) $$

For a Morton Number in the range $10^7 < 1/Mo < 10^{11}$ we can use the correlation of Klee and Treybal (1956):

$$ u_T(d) = 38.3 \rho_c^{-0.45} (\rho_c - \rho_d)^{0.58} \mu_c^{-0.11} d^{0.70}. \quad (7.29) $$

Note that the constant 38.3 is a dimensional constant and must be corrected when not using the cm-g-s system.

This terminal velocity is only valid for single droplets in an infinite medium, so we need to include correction factors for the column geometry and the effects of the surrounding droplets. The slowing factor, $k_v$, (due to the effects of the column internals) is given by Modes (1999) as:

$$ k_v = 1 - 1.1037 \left( \frac{d}{D_s - D_r} \right)^{0.12} - 0.62 \left( \frac{d}{D_s - D_r} \right)^{0.44}, \quad (7.30) $$

To allow for the effects of the droplet swarm, we use the Richardson-Zaki corre-
CHAPTER 7. 2-COMPONENT RDC

lation (Richardson and Zaki, 1954):

\[ u_r = k_w u_T (1 - \phi)^{m_{RZ}}, \]

(7.31)

with the exponent, \( m_{RZ} \), given by:

\[ m_{RZ} = 4.45 Re_T^{-0.1} - 1, \]

(7.32)

where \( Re_T \) is the Reynolds Number of the droplet at its terminal rise velocity, \( u_T \):

\[ Re_T = \frac{u_T \rho_c d}{\mu_c} \]

(7.33)

The expression for the exponent \( m_{RZ} \) given in Equation 7.32 is stated by Richardson and Zaki (1954) to be valid for a terminal Reynolds Number in the range \( 200 < Re_T < 500 \). A modification related to droplet diameter and column diameter can be made for \( Re_T \) in the range \( 1 < Re_T < 200 \), but it makes a difference of a couple of percent at most. To make calculation and simulation easier, we can regard this error as insignificant compared to other sources of error, either in experimental measurements or in the correlations themselves.

This velocity, \( u_r \), is the slip velocity, i.e., the relative velocity between the discrete and the continuous phases. To determine the absolute rise velocity of the droplets, we use the following relationship:

\[ u = u_r - \frac{Q_c}{A(1 - \phi)}, \]

(7.34)

where \( Q_c \) is the flowrate of continuous phase, \( A \) is the cross-sectional area of the column, and \( \phi \) is the holdup (volume of dispersed phase divided by total volume) in the cell.

We can now use the velocity of each particle to calculate a residence time in the cell \( (\tau_{res} = \frac{h}{u}) \) and treat inflow and outflow according to the method described in Section 3.2.1.

7.4.2 Coalescence

The model for coalescence is the corrected version of Coulaloglou and Tavlarides (1977), with the collision frequency, \( h(d_1, d_2) \), and the coalescence efficiency,
\( \lambda_K(d_1, d_2) \), given as:

\[
\begin{align*}
    h(d_1, d_2) &= \frac{c_3 \epsilon^{1/3}}{1 + \phi} (d_1 + d_2)^2 \left( d_1^{2/3} + d_2^{2/3} \right)^{1/2} \quad (7.35) \\
    \lambda_K(d_1, d_2) &= \exp \left[ -\frac{c_4 \mu_c \rho_c \epsilon}{\sigma^2 (1 + \phi)^3} \left( \frac{d_1 d_2}{d_1 + d_2} \right)^2 \right]. \quad (7.36)
\end{align*}
\]

Here, \( \epsilon \) is the power input per unit mass, and \( c_3 \) and \( c_4 \) are constants that are postulated to be universal, and which take the following values (Bapat and Tavlarides, 1985):

\[
c_3 = 1.9 \times 10^{-3} \quad c_4 = 2.0 \times 10^8. \quad (7.37)
\]

The power input per unit mass, \( \epsilon \), is given by:

\[
\epsilon = \frac{Ne \omega^3 D_r^5}{\pi D_h^2 n_c}, \quad (7.38)
\]

where \( \omega \) is the impeller rotational speed [rev/s], \( h_c \) is the height of one compartment, \( n_c \) is the number of compartments and the Newton number, \( Ne \), is a function of the system geometry. For a rotating disc contactor, we use (Simon, 2004):

\[
Ne = 2\pi \times 1.35 Re_r^{-0.4} \quad (7.39)
\]

with the characteristic Reynolds number of the rotor, \( Re_r \):

\[
Re_r = \frac{\rho_c D_r^2 \omega}{\mu_c} \quad (7.40)
\]

This is valid in the range \( 6 \times 10^3 < Re_r < 7 \times 10^4 \), so for our system, in the operating range \( 40 \text{ rpm} < \omega < 450 \text{ rpm} \). The operating conditions of \( 150 \rightarrow 300 \text{ rpm} \) all fit in this range.

Writing the coalescence kernel in terms of the droplet volumes, \( x_1 \) and \( x_2 \) gives (we revert to using \( x \) to denote volume):

\[
K(x_1, x_2) = \frac{c_1 \epsilon^{1/3}}{1 + \phi} \left( x_1^{1/3} + x_2^{1/3} \right)^2 \left( x_1^{2/9} + x_2^{2/9} \right)^{1/2} \times \exp \left( -\frac{c_2 \mu_c \rho_c \epsilon}{\sigma^2 (1 + \phi)^3} \left( \frac{(x_1 x_2)^{1/3}}{x_1^{1/3} + x_2^{1/3}} \right)^2 \right) \quad (7.41)
\]
We can construct a majorant kernel for this based on the inequalities:

\[
2 \left( x_1^{2/3} + x_2^{2/3} \right) \geq \left( x_1^{1/3} + x_2^{1/3} \right)^2 \tag{7.42}
\]

\[
\left( x_1^{1/9} + x_2^{1/9} \right) \geq \left( x_1^{2/9} + x_2^{2/9} \right)^{1/2} \tag{7.43}
\]

\[
1 \geq \exp \left( -\frac{c_2 \mu_c \rho_c \varepsilon}{\sigma^2 (1 + \phi)^3} \left( \frac{(x_1 x_2)^{1/3}}{x_1^{1/3} + x_2^{1/3}} \right)^2 \right) \tag{7.44}
\]

In fact we can make this more efficient by using a factor of \( \sqrt{2} \) instead of 2, \textit{i.e.}: (see Chapter 2 for a discussion of choosing majorant kernels)

\[
\sqrt{2} \left( x_1^{2/3} + x_2^{2/3} \right) \left( x_1^{1/9} + x_2^{1/9} \right) \geq \left( x_1^{1/3} + x_2^{1/3} \right)^2 \left( x_1^{2/9} + x_2^{2/9} \right)^{1/2} \tag{7.45}
\]

7.4.3 Breakage

Stochastic simulation is a particularly attractive method to simulate breakage in a rotating disc contactor, as models predict breakage probabilities (on transition from one cell of the contactor to the next) rather than breakage rates.

The breakage probability, \( p \), \textit{i.e.}, the probability that a particle will undergo breakage on leaving the compartment is given by \textbf{Schmidt et al.} (2003) as:

\[
\frac{p}{1 - p} = 1.2 \times 10^{-6} (\text{We}_m)^{2.88}, \tag{7.46}
\]

with the modified Weber number, \( \text{We}_m \) given by:

\[
\text{We}_m = \frac{\rho_c^{0.8} \mu_c^{0.2} d D_r^{1.6} (\omega^{1.8} - \omega_{\text{crit}}^{1.8})}{\sigma}. \tag{7.47}
\]

Here we make use of the continuous phase density, \( \rho_c \), the continuous phase viscosity, \( \mu_c \), the droplet diameter, \( d \), the rotor diameter, \( D_r \), the rotational velocity of the rotor, \( \omega \) and the critical rotor velocity, \( \omega_{\text{crit}} \). This critical rotor velocity is given by:

\[
\omega_{\text{crit}} = 2\pi \times 1.148 \left( \frac{\rho_c D_r^3}{\sigma} \right)^{-0.5} \left( \frac{d}{D_r} \right)^{-0.667}. \tag{7.48}
\]

This breakage probability is evaluated each time a droplet crosses the bound-
ary from one cell to another, to determine whether or not the droplet undergoes breakage.

Now the daughter drop probability density function is assumed to be as given by Bahmanyar and Slater (1991), so we have:

$$
\beta_{\nu}(x, y) = (\nu - 1) \left[1 - \left(\frac{y}{x}\right)^{\nu-2}\right] \frac{1}{x}.
$$

(7.49)

Here, $\nu$ is the number of daughter droplets formed and $y$ is the size of a daughter droplet formed by breakage of a droplet of size $x$. Each breakage event produces either two or three daughter droplets, with the mean number of daughter droplets, $\nu_m$ being size-dependent:

$$
\nu_m = 2 + 0.015 \left(\frac{d}{d_{\text{crit}}} - 1\right)^{2.6}
$$

(7.50)

for butylacetate/H$_2$O (Simon, 2004). $d_{\text{crit}}$ is the critical droplet diameter, related to the rotor velocity by inverting Equation (7.48):

$$
d_{\text{crit}} = D_r \left[\frac{\omega}{2\pi \times 1.148 \left(\frac{\rho_c D^3_r}{\sigma}\right)^{0.5}}\right]^{-1/0.667}
$$

(7.51)

### 7.5 Simulation Results

50 repetitions were performed using a particle number, $n$, of $2^{10} = 1024$; total simulation time was 90–100 s.

The results shown in Figure 7.1 (for equal volumetric flowrates of each phase) show that at high shaft rotation speeds, the experimentally determined results agree with those predicted by the Direct Simulation Algorithm. At lower rotation speeds however, the predicted results do not agree with the experimental measurements: the model predicts a very small amount of breakage, whereas experimental results show an increase in the number of larger droplets, indicating a significant amount of coalescence as well. The effects of the column internals are predicted by a correlation derived from single droplet experiments (Modes, 1999) and the correlation of Richardson and Zaki (1954) is then used to account for the behaviour of droplets in a swarm. CFD simulations have been performed (Vikhansky and Kraft, 2004a) that suggest a qualitatively different flow regime
Figure 7.1: Comparison of simulated and experimental results for a butylacetate-water system in a laboratory scale rotating disc contactor. Conditions: continuous phase flowrate = 100 l/min; discrete phase flowrate = 100 l/min. Dotted line = simulated inlet distribution; Solid lines = upper and lower bounds of simulated outlet distribution; • = measured inlet distribution; ▲ = measured outlet distribution.
Figure 7.2: Comparison of simulated and experimental results for a butylacetate-water system in a laboratory scale rotating disc contactor. Conditions: continuous phase flowrate = 50 l/min; discrete phase flowrate = 100 l/min. Dotted line = simulated inlet distribution; Solid lines = upper and lower bounds of simulated outlet distribution; ● = measured inlet distribution; ▲ = measured outlet distribution.
on transition from 300 rpm to 200 rpm; this might indicate that the simplified
correlations used to describe the fluid mechanics do not apply at lower shaft rota-
tional speeds. Figure 7.2 shows the comparison of predicted and measured results
for a lower continuous phase flowrate (50 l/min). In this case, there is only good
agreement between theory and experiment for the highest shaft rotation speed.

7.6 Remarks

Following the findings of the Chapter 6, the direct simulation algorithm has
been applied to a laboratory scale rotating disc contactor. The direct simulation
algorithm has been chosen in preference to the mass flow algorithm as parameters
of interest in liquid-liquid systems tend to include the Sauter mean diameter, $d_{32}$,
and other “less than one” moments. By employing a simulation algorithm that
resolves the whole particle size distribution, it becomes straightforward to validate
(or otherwise) theoretical models. In this case, the model uses single droplet
behaviour to predict droplet behaviour in a swarm. For high shaft rotational
speeds, the model provides an accurate prediction of the experimental results.
For lower rotational speeds however, the simulation and experimental results
differ significantly, suggesting deficiencies in the model; this is backed up by CFD
simulations (Vikhansky and Kraft, 2004a), suggesting qualitatively different flow
regimes at the two rotational speeds.
Chapter 8

Simulation of a 3-Component RDC

8.1 Mass Transfer to/from a Sphere

8.1.1 Mass Transfer Resistance In The Continuous Phase

If mass transfer in the discrete phase can be considered to be sufficiently fast that the resistance is controlled by the continuous phase, a correlation in terms of dimensionless groups is normally used. Skelland (1974, pp.276-277) records correlations from a number of different sources of the form $Sh = 2 + k' Re^{1/2} Sc^{1/3}$ or $Sh = k' Re^{1/2} Sc^{1/3}$, where $0.33 \leq k' \leq 0.95$. There can also be a slight variation in the Reynolds number exponent in the second expression. The Schmidt number exponent of $1/3$ represents the theoretical limit of an immobile droplet interface ($\mu_d/\mu_c \to \infty$) in potential flow ($Re \to \infty$). Alternatively, the theoretical calculations of Boussnesq (1905) give $Sh = (4/\pi)^{1/2} Re^{1/2} Sc^{1/2}$ in the limit of a clean liquid droplet with a mobile interface ($\mu_d/\mu_c \to 0$).

Levins and Glastonbury (1972) proposed a semi-empirical correlation for mass transfer to spherical particles suspended in a stirred vessel:

$$Sh = 2 + 0.47 \left( \frac{d^{4/3} \varepsilon^{1/3} \rho_c}{\mu_c} \right)^{0.62} \left( \frac{D_S}{D_C} \right)^{0.17} Sc^{0.36}$$  \hspace{1cm} (8.1)

where $D_S$ is the stirrer diameter and $D_C$ is the cell diameter.
Calderbank and Moo-Young (1961) give the following relationship in the case when mass transfer is due to turbulence in the surrounding fluid:

\[ k_{e,c}S_c^{2/3} = 0.13 \left( \frac{\varepsilon \mu_c}{\rho_c} \right)^{1/4}. \]  

\[ (8.2) \]

### 8.1.2 Mass Transfer Resistance In The Discrete Phase

The first solution of mass transfer within a spherical particle was given by Newman (1931) for the case of pure diffusion:

\[ \frac{Y_{d,e} - \bar{Y}_d(t)}{Y_{d,e} - Y_{d,0}} = \frac{6}{\pi^2} \sum_{n=1}^{\infty} \frac{\exp[-(n\pi)^2 Fo]}{n^2}, \]  

\[ (8.3) \]

where \( \bar{Y}_d(t) \) indicates that the mass fraction of solute in the droplet, \( Y_d \), has been averaged over the droplet volume, \( Y_{d,e} \) is the equilibrium mass fraction, and \( Y_{d,0} \) is the initial value. The Fourier number, \( Fo \), is a dimensionless time, dependent on the molecular diffusion rate within the particle, \( D_d \), and the particle diameter, \( d \):

\[ Fo = \frac{4D_d t}{d^2}. \]  

\[ (8.4) \]

Note that the solution for the evolution of the solute mass fraction as given in equation 8.3 can be inefficient to calculate, especially for small values of \( t \) as would be required for stochastic simulation. In this case, we can replace the expression in equation 8.3 with the following approximation (Henschke and Pfennig, 1999):

\[ \frac{Y_{d,e} - \bar{Y}_d(t)}{Y_{d,e} - Y_{d,0}} = \begin{cases} 
1 - 6\sqrt{\frac{Fo}{\pi}} + 3Fo & : Fo \leq 0.15 \\
\frac{6}{\pi^2} \exp(-\pi^2 Fo) & : Fo > 0.15.
\end{cases} \]  

\[ (8.5) \]

Alternatively, Vermeulen used the approximation:

\[ \frac{Y_{d,e} - \bar{Y}_d(t)}{Y_{d,e} - Y_{d,0}} = 1 - \left[ 1 - \exp(-\pi^2 Fo) \right]^{1/2} \]  

\[ (8.6) \]

Equation 8.3 can be used to describe unsteady internal mass transfer in the case where the mass transfer rate is enhanced by internal circulation; now an enhanced diffusion coefficient, \( D_{eff} = R D_d \), replaces \( D_d \). One of the earliest values for \( R \) came from the work of Kronig and Brink (1950), who studied internal circu-
lation at low Reynolds numbers and zero interfacial tension; an approximation to their infinite series sum corresponds to the approximation given in equation 8.6 and gives a value of $R = 2.25$. Alternatively, $R$ can be determined empirically; see e.g. Calderbank and Korchinski (1956) ($1.8 < R < 3.3$), Boyadzhiev et al. (1969) ($R \propto Re^2$), Steiner (1986) ($R = f(Re, Sc)$).

Henschke and Pfennig (1999) used equation 8.3 with a modified diffusion coefficient to take into account diffusion and turbulent mixing:

$$D_{eff} = D_d + \varepsilon_m^\sigma$$  \hspace{1cm} (8.7)

Here, $\varepsilon_m^\sigma$ combines the enhancement in mass transfer due to both mass-transfer induced turbulence ($\varepsilon^\sigma$) and hydrodynamic turbulence ($\varepsilon_m$):

$$\varepsilon_m^\sigma = \varepsilon^\sigma + \varepsilon_m = \frac{u_r d}{C_{IP} \left(1 + \frac{\mu_d}{\mu_c}\right)}$$  \hspace{1cm} (8.8)

The adjustable parameter $C_{IP}$ is called the instability constant, and characterises the instabilities at the interface specific to the system. For $C_{IP} \rightarrow \infty$, the limit of purely diffusive mass transfer is obtained, and $C_{IP} \rightarrow 0$ gives the limit of infinitely fast turbulent mixing. According to the often-used model of Handlos and Baron (1957), $C_{IP}$ takes the value of 2048; this model is applicable for the case of non-oscillating droplets, i.e. in the high–Re case (Zhang et al., 1985). A simplification of this model (by taking only the first term in the infinite sum) gives:

$$k_d = 0.00375 u_r \frac{1}{1 + \mu_d/\mu_c}.$$  \hspace{1cm} (8.9)

This is only accurate for long contact times (Brunson and Wellek, 1970). A modification can be made for shorter timescales:

$$k_d = 0.00365 u_r \frac{1}{1 + \mu_d/\mu_c} + 0.3 \frac{D}{t_d}.$$  \hspace{1cm} (8.10)

Many more models for discrete and continuous phase mass transfer resistances exist; most are empirical in nature and are not listed here. Kumar and Hartland (1999) contains a good summary of a large number of models.
8.1.3 Mass Transfer Resistance In Both Phases

The preceding equations governing internal mass transfer assume an infinite mass transfer rate in the continuous phase. Steiner (1986) combines both internal and external mass transfer by using an empirical correlation for the external mass transfer coefficient and then uses this mass transfer rate as a boundary condition in Newman’s internal mass transfer equation. Now, the mass fraction of solute in the droplet is given by:

\[
\frac{Y_e - \bar{Y}(t)}{Y_e - Y_0} = \sum_{n=1}^{\infty} \frac{6L^2 \exp[-\beta_n^2 Fo]}{\beta_n^2[\beta_n^2 + L(L-1)]}
\] (8.11)

where \(\beta_n\) and \(L\) are given by:

\[
\beta_n \cot \beta_n + L - 1 = 0
\]

\[
L = \frac{Sh'_c}{2}
\]

\[
Sh'_c = \frac{\rho_c D_c}{\rho_d D_d} Sh_c
\]

\[
\frac{Sh_c - Sh_{cr}}{Sh_{ca} - Sh_{cr}} = 1 - \exp[-0.418 \times 10^{-3} Pe^{0.42}]
\]

\[
Sh_{ca} = \frac{2}{\sqrt{\pi}} \frac{Re^{1/2} Sc^{1/2}}{Re \times Sc^{1/3}}
\]

\[
Sh_{cr} = 2.43 + 0.775 Re^{1/2} Sc^{1/3} + 0.0103 Re Sc^{1/3}.
\] (8.12)

In the last of these expressions, the value of 2.43 can be replaced by its theoretical value of 2, or even left out completely, as it does not contribute significantly to the value of \(Sh_{cr}\).

A more common (and simpler) approach than the one presented above is to model the mass transfer resistances of the discrete and continuous phases separately, and then combine them in the familiar way (Kay and Nedderman, 1985):

\[
k_{o,d} = \left( \frac{1}{k_d} + \frac{\kappa}{k_c} \right)^{-1}.
\] (8.13)

8.1.4 Which Phase?

Dang et al. (1970) considered the problem of the Quasi-Steady Assumption for Mass Transfer to spherical droplets. They found that the quasi-steady state
assumption is satisfied if:

$$\kappa Pe^{1/2} \gg 1,$$

(8.14)

where $\kappa$ is the solute equilibrium distribution constant – the ratio of the equilibrium concentration of the solute in the dispersed phase to that in the continuous phase:

$$\kappa = \frac{c_{d,e}}{c_{c,e}}.$$  

(8.15)

The Peclet Number, $Pe$, has two different forms for two different cases:

$$Pe = \begin{cases} \frac{1}{4} \frac{\mu_c - \mu_d}{\mu_c + \mu_d} \frac{du}{D_c} : \text{small Reynolds Number} \\ \frac{3}{4} \frac{du}{D_c} : \text{potential flow (high } Re \text{).} \end{cases}$$

(8.16)

In these expressions, $\mu_c$ and $\mu_d$ are the continuous and discrete phase viscosities, $d$ is the diameter of a droplet, $u$ is its translational velocity and $D_c$ is the molecular diffusivity in the continuous phase.

Further to this quasi-steady state criterion is the assumption that the mass transfer resistance is confined to the continuous phase. This assumption is found to hold when:

$$\kappa \frac{D_d}{D_c} Pe^{-1/2} \gg 4 \times 10^{-2}$$

(8.17)

For liquid-liquid dispersions (i.e., similar diffusivities), this imposes the practical restriction that $\kappa$ must be large, i.e., the equilibrium concentration of the solute is much higher in the dispersed phase than it is in the continuous phase.

### 8.2 Multi-dimensional Stochastic Simulation

#### 8.2.1 Stochastic Particle Array

For a given physical situation, there will be a specified vector of particle properties that need to be stored. In the case of a three component liquid-liquid system, this could be a two-dimensional vector comprising droplet mass and solute mass fraction (this makes the simplifying assumption that one component is confined entirely to the continuous phase, one component is confined entirely to the discrete phase, and the third component, the solute, is transferred between the two). To extend single-dimensional (size dependent only) stochastic simulation to the
multi-dimensional case, we need to use the following approach:

Our stochastic particle system, for an $S$-dimensional system, now looks like:

$$c(t, \vec{x}) \approx U^N(t, \vec{x}) = p(\vec{x}) = \frac{1}{N} \sum_{i=1}^{n} \delta(\vec{x} - \vec{x}_i)$$

(8.18)

where $\vec{x}$ denotes the vector of particle properties of interest:

$$\vec{x} = (x_1, x_2, \ldots, x_S)^T,$$

(8.19)

the vector of properties for the stochastic particle at index $i$ is:

$$\vec{x}_i = (x_{1,i}, x_{2,i}, \ldots, x_{S,i})^T,$$

(8.20)

and the shorthand notation $\delta(\vec{x} - \vec{x}_i)$ indicates:

$$\delta(\vec{x} - \vec{x}_i) \equiv \prod_{s=1}^{S} \delta(x_s - x_{s,i})$$

(8.21)

We can now predict any functional, $\Phi(t)$, of the particle system:

$$\Phi(t) = \int_{0}^{\infty} \phi(\vec{x}) c(t, \vec{x}) d\vec{x}$$

$$= \int_{x_1=0}^{\infty} \ldots \int_{x_S=0}^{\infty} \phi(x_1, \ldots, x_S)c(t, x_1, \ldots, x_S)dx_1 \ldots dx_S$$

$$\approx \int_{x_1=0}^{\infty} \ldots \int_{x_S=0}^{\infty} \phi(x_1, \ldots, x_S)\frac{1}{N} \sum_{i=1}^{n} \prod_{s=1}^{S} \delta(x_s - x_{s,i}) dx_1 \ldots dx_S$$

$$\approx \frac{1}{N} \sum_{i=1}^{n} \int_{x_1=0}^{\infty} \ldots \int_{x_S=0}^{\infty} \phi(x_1, \ldots, x_S) \prod_{s=1}^{S} \delta(x_s - x_{s,i}) dx_1 \ldots dx_S$$

$$\approx \frac{1}{N} \sum_{i=1}^{n} \phi(x_{1,i}, \ldots, x_{S,i}) \approx \frac{1}{N} \sum_{i=1}^{n} \phi(\vec{x}_i)$$

(8.22)

In order that our stochastic simulation correctly predicts the behaviour of the real system, we need to formulate conservation laws for each jump process that occurs. Recall that in the simple one-dimensional coagulation case, the jump process looks like:

$$[x_1] + [x_2] \rightarrow [x_1 + x_2]$$

(8.23)
i.e., the particles of sizes $x_1$ and $x_2$ are removed and replaced by one of size $x_1 + x_2$. In the two-dimensional case of a three-component liquid-liquid system, the jump process for coagulation can be simply formulated by mass conservation as:

$$\begin{bmatrix} m_1 \\ Y_1 \end{bmatrix} + \begin{bmatrix} m_2 \\ Y_2 \end{bmatrix} \rightarrow \begin{bmatrix} m_1 + m_2 \\ (m_1Y_1 + m_2Y_2)/(m_1 + m_2) \end{bmatrix} \quad (8.24)$$

where we use $m_i$ to denote the mass of the $i$th particle in the array. It may also be convenient to store other properties of each particle (e.g. density, volume, diameter etc.) and update them at each step, but the system still has only two independent dimensions.

### 8.3 Stochastic Treatment of Mass Transfer

According to the models commonly employed to describe it, mass transfer is a continuous process. This makes it different to those processes already examined: coalescence, breakage, inflow and outflow, which are discrete, and therefore ideally suited to stochastic simulation by means of waiting times and discrete jumps. If we wish to stochastically simulate mass transfer then we need to get around this problem, either by constructing a model that treats mass transfer as a discrete process, or by carefully choosing when to update the system according to the continuous model.

If the mass transfer resistance is confined to the continuous phase, then we can use the following instantaneous mass balance:

$$\frac{\pi}{6}d^3 \frac{dY_d}{dt} = -k\pi d^2 \left( \frac{Y_d}{\kappa} - Y_c \right) \quad (8.25)$$

Note that we are using mass fraction (rather than concentration) as the variable of interest here. We do so to make formulation of conservation laws (based on mass conservation) easier. Strictly speaking, the above model makes the assumption that droplet and continuous phase densities are constant, which is reasonable for small changes in mass fraction.
For convenience, we can introduce the following definitions:

characteristic time for mass transfer: \( t_k = \frac{\kappa d}{6k} \) \hspace{1cm} (8.26)

equilibrium solute concentration in the droplet: \( Y_{d,e} = \kappa Y_c \) \hspace{1cm} (8.27)

so the mass transfer model becomes:

\[
\frac{dY_d}{dt} = -\frac{1}{t_k} (Y_d - Y_{d,e})
\] \hspace{1cm} (8.28)

In a stochastic simulation, this equation can be solved in one of three ways:

### 8.3.1 Continuous Treatment

Firstly, it can be solved for each interval of quiescence, \( \tau \), and the mass fraction in each droplet can be updated after each time step according to:

\[
Y_{d,i} \rightarrow \kappa Y_c + (Y_{d,0} - \kappa Y_c) \exp \left( -\frac{\tau}{t_{k,i}} \right)
\] \hspace{1cm} (8.29)

where \( y_{d,0} \) is the solute mass fraction in the droplet at the beginning of the interval of quiescence. It is also necessary to update the continuous phase mass fraction. The amount of solute transferred from a single droplet to the continuous phase during the interval of quiescence is given by:

\[
\Delta m_i = \rho_{d,i} m_i (Y_{d,i} - Y_{d,e}) \left[ 1 - \exp \left( -\frac{\tau}{t_{k,i}} \right) \right]
\] \hspace{1cm} (8.30)

This is incorporated in a material balance over the continuous phase, including in- and out-flow, to give the new continuous phase mass fraction:

\[
Y_c \rightarrow Y_c + \frac{1}{\rho_c V_c} \left[ \sum_{i=1}^{n} m_i + \frac{V_c \tau}{\tau_{res}} (Y_{c,f} - Y_c) \right]
\] \hspace{1cm} (8.31)

where the volume of the continuous phase, \( V_c \), is:

\[
V_c = (1 - \phi)V = \frac{1 - \phi}{\phi} V_d
\] \hspace{1cm} (8.32)
This method has been used by (e.g. Bapat et al., 1983; Bapat and Tavlarides, 1985)

If there is internal mass transfer resistance in addition to (or instead of) the external resistance, then we can simply use the same method as above, but using equation 8.11 (or 8.3) to work out \( m_i \) in place of equation 8.30.

### 8.3.2 Discrete Treatment

The second method of dealing with a continuous transfer of material from one phase to another is to approximate it as a set of discrete changes in concentration and treat it as a stochastic jump process with its own stochastic time step parameter.

We can now rewrite equation 8.28 in terms of a population balance:

\[
\frac{\partial c(t, m, Y_d)}{\partial t} = \frac{1}{t} \frac{\partial}{\partial Y_d} ((Y_d - Y_{d,e})c(t, m, Y_d)) + \ldots
\]  
(8.33)

For convenience, in this equation we have left off the other terms in the population balance equation, describing coalescence, breakage and throughflow. This describes the rate of change with respect to time of the number concentration at time \( t \) of droplets of mass \( m \) and solute mass fraction \( Y_d \). For our discrete approach to this continuous change in mass fraction, we make the following approximation (again, for convenience, we only examine the mass transfer terms):

\[
\frac{\partial c(t, m, Y_d)}{\partial t} = \frac{1}{t} \rho_k \sum_{i=1}^{n} \left[ \rho_\Delta(Y_{d,i})c(t, m, Y_{d,i}) \right] + \ldots
\]  
(8.34)

with the function \( \rho_\Delta(Y_{d,i}) \) given by:

\[
\rho_\Delta(Y_{d,i}) = \begin{cases} 
\frac{1}{\Delta Y} |Y_d - Y_{d,e}| & : \text{if } |Y_{d,e} - Y_d| > \Delta Y \\
0 & : \text{otherwise}.
\end{cases}
\]  
(8.35)

Thus we have for the contribution to the time step parameter due to mass transfer:

\[
\rho_k = \sum_{i=1}^{n} \frac{1}{t_{k,i}} \rho_\Delta(Y_{d,i})
\]  
(8.36)

where the subscript \( i \) denotes the particle with index \( i \) in the stochastic particle
array. Note that the characteristic time for mass transfer, $t_k$, is also denoted with a particle index in the above expression because it depends on the particle diameter.

Note here that the equilibrium concentration, $Y_{d,e} = \kappa Y_c$, varies with the concentration of solute in the continuous phase. To avoid having to update $\rho_k$ for every particle every time the continuous phase concentration changes, a simple majorant can be introduced, \textit{i.e.}:

$$\hat{\rho}_k = \sum_{i=1}^{n} \frac{1}{t_{k,i}} \hat{\rho}_{\Delta,i}(Y_{d,i}),$$

(8.37)

where

$$\hat{\rho}_{\Delta} = \frac{Y_d}{\Delta Y} \geq \rho_{\Delta}$$

(8.38)

Because it is true that $\hat{\rho}_{\Delta,i} \geq \rho_{\Delta,i}$ for all $i$, it must also be true that $\hat{\rho}_k \geq \rho_k$. Now, the majorant can be used and fictitious jumps can be introduced for efficient simulation.

Upon each time step when mass transfer is the chosen process, the particle is chosen according to:

$$\frac{1}{t_{k,i}} \hat{\rho}_{\Delta,i}(Y_{d,i}) \sum_{i=1}^{n} \frac{1}{t_{k,i}} \hat{\rho}_{\Delta,i}(Y_{d,i})$$

(8.39)

and with probability:

$$\frac{\rho_{\Delta,i}}{\hat{\rho}_{\Delta,i}},$$

(8.40)

particle $i$ has its solute concentration updated to:

$$Y_{d,i} \rightarrow Y_{d,i} - \Delta Y$$

(8.41)

The continuous phase solute concentration also must be updated to:

$$Y_c \rightarrow Y_c + \frac{\rho_{d,i} m_i \Delta Y}{V_c \rho_c}$$

(8.42)

If the internal mass transfer resistance cannot be neglected, then the discrete treatment must be amended, due to the non-constant rate of mass transfer.
8.3.3 Process Deferment

Stochastic simulation of continuous processes can also be done using the Linear Process Deferment Algorithm (Patterson et al., 2006). This is based on the operator splitting algorithm method of simulating continuous processes (e.g. Bhave and Kraft, 2004; Vikhansky and Kraft, 2004b). The application of this method to our situation is as follows:

Coalescence, throughflow and breakage are processes that depend primarily on droplet size. There is dependence of fluid properties on the concentration of solute, but this is small compared to the size dependence. Mass transfer is a process that depends on droplet size and concentration, so if these two variables remain unchanged over a (short) period of time, we can use an equation such as 8.3 or 8.11 to update the concentration (or mass fraction) of solute in a droplet. The events (apart from mass transfer) that change the identity of a particle, and therefore its size and/or concentration are coalescence and breakage, so we propose that it is only upon a coalescence or breakage step that a mass transfer update needs to be performed. If we keep track of the age of each particle (or more efficiently, its inception time, $t_i$, such that the age is the inception time subtracted from the current time), then this age can be substituted into an equation such as 8.3 to update the concentration immediately before a coagulation or breakage step takes place. The new particle(s) formed then have an inception time set to the current time (age is effectively set to zero) and the process is repeated. In a multi-cell system we should also take care whenever a particle is introduced from another cell. In this case, the identity of the droplet is unchanged, but the nature of the surrounding fluid may be substantially different in the two cells. We can therefore consider a jump from one cell to the next to be identity destroying so we should also perform the mass transfer update before introducing the particle into the new cell.

To summarise the simulation algorithm in this case:

1. Initialise the particle ensemble, with time $t = 0$ and $t_i = 0$ for all particles.

2. Generate a time step, $\tau$, with parameter determined by the rates of coalescence and of in- and out-flow.

3. Set time $t = t + \tau$ and choose either:
(a) Coalescence:
   i. Choose two particles for coalescence and perform mass transfer (at age $t - t_i$) for each.
   ii. Update continuous phase properties.
   iii. Perform coalescence step.
   iv. Set $t_i = t$ for newly formed particle.
   v. Duplicate a particle and update normalisation parameter.

(b) Outflow:
   i. Remove outflowing particle.
   ii. Duplicate a particle and update normalisation parameter.

(c) Inflow:
   i. Perform mass transfer update (at age $t = t - t_i$) for inflowing particle.
   ii. Update continuous phase properties.
   iii. If breakage is to be performed, calculate daughter particle properties.
   iv. Set $t_i = t$ for newly introduced particle(s).
   v. Remove particle(s) and update normalisation parameter.

4. Return to step 2.

8.4 Algorithm Testing

To give an idea of which of these three methods is most appropriate, we simulate a simple case, in which coagulation and mass transfer are the only events occurring. The following model is used:

\begin{align*}
K(m_1,m_2) &= 1 \\
\frac{dY}{dt} &= -kY, \quad k = 0.015 \\
Y(0) &= 0.2
\end{align*}
Thus, in this simple case, both mass transfer and coalescence rates are independent of size and a simple analytical solution can be derived for the mass fraction at any time. Table 8.1 shows the run times (on an 866Mhz PC) for each of the three stochastic algorithms; Figure 8.1 shows the results achieved from each algorithm, using $n = 512$. The continuous algorithm shows excellent accuracy (Figures 8.1(a), 8.1(b)) but the run time increases as $n^2$. The run time for the discrete algorithm shows better scaling characteristics (scales as $n$), but does not show as good accuracy. Even in this very simple case, the discrete jump, $\Delta Y$, has to be very small to achieve decent accuracy (Figures 8.1(c), 8.1(d)). The deferred algorithm combines the best characteristics of the first two algorithms, with the run time scaling as $n$, and the accuracy being very good. The results for different values of $\Delta t$ are exactly the same, hence there is only one set of simulated results on each of Figure 8.1(e) and Figure 8.1(f).

**Table 8.1: Comparison of run times for three stochastic algorithms.**

<table>
<thead>
<tr>
<th>$n$</th>
<th>$t_{sr}/s$</th>
<th>$\Delta Y$</th>
<th>$t_{sr}/s$</th>
<th>$\Delta t$</th>
<th>$t_{sr}/s$</th>
</tr>
</thead>
<tbody>
<tr>
<td>256</td>
<td>0.385</td>
<td>0.0001</td>
<td>0.358</td>
<td>0.01</td>
<td>1.40</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.001</td>
<td>0.0384</td>
<td>0.1</td>
<td>0.153</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.01</td>
<td>0.00875</td>
<td>1</td>
<td>0.0222</td>
</tr>
<tr>
<td>512</td>
<td>1.66</td>
<td>0.0001</td>
<td>0.786</td>
<td>0.01</td>
<td>3.06</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.001</td>
<td>0.0841</td>
<td>0.1</td>
<td>0.328</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.01</td>
<td>0.0159</td>
<td>1</td>
<td>0.0475</td>
</tr>
<tr>
<td>1024</td>
<td>6.66</td>
<td>0.0001</td>
<td>1.59</td>
<td>0.01</td>
<td>6.04</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.001</td>
<td>0.172</td>
<td>0.1</td>
<td>0.636</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.01</td>
<td>0.0297</td>
<td>1</td>
<td>0.0956</td>
</tr>
</tbody>
</table>

Looking more closely at the behaviour of the discrete algorithm, we can see that the results are normally distributed about a mean value equal to the analytical solution, as illustrated in Figure 8.2. As the simulation progresses, the simulated results get more spread out (standard deviation increases) and then narrow again (standard deviation decreases) as the equilibrium value is approached. This suggests that the discrete algorithm is unsuited for a transient response, but may be effective in simulating steady state behaviour. The variation of the standard deviation with the simulation parameter $\Delta Y$ is shown in Figure 8.3, showing that
Figure 8.1: Test case simulation results for three different algorithms.
the standard deviation varies as $\Delta Y^{1/2}$. As $\Delta Y$ is decreased, the standard deviation decreases, but the simulation time (Table 8.1) increases faster than this, confirming that this is an inefficient method of simulation.

<table>
<thead>
<tr>
<th>$\Delta Y$</th>
<th>$t$</th>
<th>mean</th>
<th>st. dev.</th>
<th>err. $\times 10^5$</th>
<th>st. dev.</th>
<th>err. $\times 10^5$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$0.01$</td>
<td>10</td>
<td>0.172</td>
<td>0.0136</td>
<td>1350</td>
<td>0.00419</td>
<td>39.1</td>
</tr>
<tr>
<td></td>
<td>20</td>
<td>0.148</td>
<td>0.0169</td>
<td>966</td>
<td>0.00531</td>
<td>30.6</td>
</tr>
<tr>
<td></td>
<td>30</td>
<td>0.128</td>
<td>0.0186</td>
<td>460</td>
<td>0.00577</td>
<td>27.3</td>
</tr>
<tr>
<td></td>
<td>40</td>
<td>0.110</td>
<td>0.0193</td>
<td>213</td>
<td>0.00602</td>
<td>17.7</td>
</tr>
<tr>
<td></td>
<td>50</td>
<td>0.0945</td>
<td>0.0195</td>
<td>112</td>
<td>0.00608</td>
<td>15.6</td>
</tr>
<tr>
<td>$0.001$</td>
<td>60</td>
<td>0.0813</td>
<td>0.0191</td>
<td>81.2</td>
<td>0.00603</td>
<td>14.3</td>
</tr>
<tr>
<td></td>
<td>70</td>
<td>0.0700</td>
<td>0.0186</td>
<td>173</td>
<td>0.00587</td>
<td>16.7</td>
</tr>
<tr>
<td></td>
<td>80</td>
<td>0.0602</td>
<td>0.0180</td>
<td>193</td>
<td>0.00568</td>
<td>17.9</td>
</tr>
<tr>
<td></td>
<td>90</td>
<td>0.0518</td>
<td>0.0173</td>
<td>415</td>
<td>0.00547</td>
<td>11.2</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>0.0446</td>
<td>0.0164</td>
<td>1020</td>
<td>0.00520</td>
<td>40.0</td>
</tr>
</tbody>
</table>

8.5 Simulation of mass transfer in a rotating disc contactor

We now wish to validate our stochastic simulation method for the multi-dimensional case; we do so by comparison with the bivariate quadrature method of moments technique (Attarakih et al., 2006). Attarakih et al. (2006) modelled a rotating disc contactor with breakage, coalescence and mass transfer occurring. We use the same rise velocity, breakage and coalescence models as described in Sections 7.4.1–7.4.3. The situation modelled is a butylacetate-water system, with acetone as the solute; we use the correlations given in Bart et al. (1994) for the physical properties.

Attarakih et al. (2006) used two mass transfer models in their work; those of Handlos and Baron (1957) as applied by Weinstein et al. (1997), and of Kumar and Hartland (1999). As they find very little difference between the two models in their simulations, we examine only the first of these models, using Equation (8.10) for the discrete phase mass transfer resistance and the relationship measured by
Figure 8.2: Comparison of simulated results (dashed lines) and assumed shape normal distribution (solid). The parameters for the normal distribution are calculated by using the analytical solution for the mean and the method of least squares to determine the standard deviation.
Figure 8.3: Convergence of the standard deviation with $\Delta Y$ at various simulation times. The solid lines have a slope of 0.5.

Heertjes et al. (1954) (a form of the relationships listed by Skelland (1974)) for the discrete phase mass transfer resistance:

$$k_c = 0.83 \sqrt{\frac{u_r D_c}{d}}.$$ (8.46)

Simulation is performed with five stochastic populations representing the droplets in the five compartments, each with a particle number of $n = 2^7$. The LPDA method (Chapter 8.3.3) is used to implement the continuous mass transfer models. Fifty repetitions are performed, for a total cpu time (on a 2 GHz desktop PC) of about 30 s. Good agreement is found, with cpu times comparable to those of Attarakih et al. (2006), despite the different approach to implementing the physical models.
Figure 8.4: Simulation of mass transfer in a rotating disc contactor. Comparison of simulated results (dotted lines) with the results of Attarakih et al. (2006) (solid lines).
Chapter 9

Further Work

9.1 Scope of the Chapter

Having established that stochastic solution methods are effective for solving multi-dimensional population balance equations, we now wish to demonstrate how they can be useful in developing models, not just in solving them. Each jump in the stochastic model is a representation of a real life process that occurs, so implementing them is a very intuitive process. In addition, the solution method enables comparison of any property of interest to be extracted and compared with experimental results. Thus, experiments can be designed in order to validate (or otherwise) any proposed physical models.

A simple multi-dimensional population balance is proposed that incorporates physical observations of growth of particles in wet powder agglomeration processes as a function of size, surface liquid and deformability. A simple function for the probability that a collision results in coagulation is used, categorising particles as small/large, wet/dry and soft/hard according to physical properties such as solid volume, liquid content, porosity and surface area. Depending on the deformability of coagulating particles, we propose conservation rules for the above properties when a coagulation step occurs. In addition, we characterise the aggregates with a fractal dimension, enabling us to relate surface area to solid volume and pore volume, thus reducing the problem to three independent variables.

At this stage, a full validation of the proposed model by comparison with experimental data is not attempted; instead, it is proposed that stochastic solution
of multi-dimensional population balance models will be an effective tool in future modelling of wet powder agglomeration.

9.2 Wet Powder Agglomeration

Agglomeration is a size enlargement process in which small particles are gathered together to form larger relatively permanent aggregates. This may be achieved by spraying liquid binder onto a dry powder bed inside a process vessel such as a rotating drum or high shear mixer. After more than fifty years of research the processes of agglomeration have been identified as nucleation, growth and breakage and within each there are various regimes. However it is not always known which mechanism is operating, most likely because there is very little knowledge of the forces and inter-particle collisions occurring in agglomeration equipment. Thus it is difficult to predict the outcome of agglomeration and there is no formal methodology for design or scale-up.

The population balance equation describes changes in a population in which each member has one or more characteristic properties. In agglomeration the use of population balance models is widespread (Liu and Litster, 2002; Sastry and Fuerstenau, 1973; Adetayo and Ennis, 1997, 2000) however the most important parameter for the model, the coalescence rate kernel, is nearly always empirically based. Size, surface liquid and deformability are known to influence the success of collisions between agglomerates, so a multi-dimensional population balance is essential for describing agglomeration (Iveson, 2002).

Traditionally, many different mechanisms have been proposed to describe the agglomeration process (Sastry and Fuerstenau, 1973). These are mainly subsets of a smaller number of processes and more recent descriptions of the mechanisms involved reflect this (Iveson et al., 2001a; Ennis and Litster, 1997)

9.2.1 Multi-Dimensional Population Balance Model

Wauters (2001) proposes a multi-dimensional population balance model for the aggregation process based on three parameters: $v_s$, the volume of solid in a particle, $v_l$, the volume of liquid in a particle, and $v_a$, the volume of air in a particle. The model is formulated based on observations of the granulation process. Small granules are picked up by wet spots on large granules. When the small granules
have disappeared the large granules become surface wet and can coalesce with each other. Thus, using the following definitions of “small”, “big”, “wet” and “dry”:

\[
\sqrt[3]{\frac{6(v_s + v_l + v_a)}{\pi}} < 500 \mu m : \text{“small”}
\]

\[
\sqrt[3]{\frac{6(v_s + v_l + v_a)}{\pi}} > 500 \mu m : \text{“big”}
\]

\[
\frac{v_l}{v_l + v_a} < 0.85 : \text{“dry”}
\]

\[
\frac{v_l}{v_l + v_a} > 0.85 : \text{“wet”}
\]

we can construct a table to summarise the probability of coalescence as a function of the particle properties. This coalescence function is denoted \( f_{i,j} \):

**Table 9.1:** Probability of coalescence as function of properties of granules \( i \) and \( j \).

<table>
<thead>
<tr>
<th>( i ) ( \downarrow ) ( j ) ( \rightarrow )</th>
<th>big &amp; wet</th>
<th>big &amp; dry</th>
<th>small &amp; wet</th>
<th>small &amp; dry</th>
</tr>
</thead>
<tbody>
<tr>
<td>big &amp; wet</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>big &amp; dry</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>small &amp; wet</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>small &amp; dry</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

The form of the coalescence kernel is now:

\[
K(v_{s,i}, v_{l,i}, v_{a,i}, v_{s,j}, v_{l,j}, v_{a,j}) = K_0 f_{i,j}(v_{s,i}, v_{l,i}, v_{a,i}, v_{s,j}, v_{l,j}, v_{a,j}),
\] (9.1)

where \( f_{i,j} \) is as described in Table 9.1 and \( K_0 \) is a size-independent part (although this formulation can be generalised such that \( K_0 \) is a function of time).

Wauters also includes a compaction (porosity reduction) mechanism in his model. First order porosity reduction was first proposed by Iveson *et al.* (1996) as:

\[
\frac{d\varepsilon}{dN} = -k(\varepsilon - \varepsilon_{\text{min}}).
\] (9.2)

Here, \( N \ [\text{rev}] \) is the number of drum revolutions, \( \varepsilon_{\text{min}} \ [\text{–}] \) is the final (equilibrium) granule porosity and \( k \ [\text{rev}^{-1}] \) is a rate constant. The expression is easily converted to a time derivative by multiplying the rate constant by the drum
rotation speed.

Wauters considers the porosity in terms of $v_s$, $v_l$ and $v_a$:

$$\varepsilon = \frac{v_l + v_a}{v_s + v_l + v_a}$$  \hspace{1cm} (9.3)

and assumes that porosity occurs by reduction of the air volume in a particle, with the solid and liquid volumes remaining constant. Thus:

$$\frac{d\varepsilon}{dt} = \frac{dv_a}{dt} \frac{v_s}{(v_s + v_l + v_a)^2}$$  \hspace{1cm} (9.4)

A problem with this approach is that under certain conditions (depending on the initial liquid content and the final porosity), the model predicts that the air volume can become negative. The model also necessarily predicts that total volume, $v$, is conserved after a coagulation step. It has been observed (Forrest, 2003) that under some regimes, coagulation without rearrangement produces a raspberry-like structure that increases the pore volume (and therefore the total volume) of the aggregate.

We therefore modify the approach of Wauters slightly to consider the pore volume, $v_p$, and write the porosity as:

$$\varepsilon = \frac{v_p}{v_s + v_p}$$  \hspace{1cm} (9.5)

Compaction proceeds by reduction of the pore volume (instead of the air volume), according to:

$$\frac{dv_p}{dt} = \frac{d\varepsilon}{dt} \frac{(v_s + v_p)^2}{v_s} = -k' \left( \frac{v_p(v_s + v_p)}{v_s} - \frac{\varepsilon_{\text{min}}(v_s + v_p)^2}{v_s} \right)$$  \hspace{1cm} (9.6)

We now assume that any liquid will occupy the pores of the particle until, above a certain liquid content, the liquid has to cover the surface of the particle, and the particle is considered “wet” for the purposes of the coagulation model. The point at which the particle becomes surface wet is taken to be when the liquid content rises above 85% of the pore volume, in line with the coagulation model.

For our model we now have to keep track of three independent quantities: the solid volume, $v_s$, the liquid volume, $v_l$ and the pore volume, $v_p$. These three quantities define the remaining parameters of interest, \textit{e.g.} the volume of air in
the particle:

\[ v_a = \begin{cases} 
  v_p - v_l & : v_l < 0.85v_p \\
  0.15v_p & : v_l > 0.85v_p,
\end{cases} \tag{9.7} \]

or the total particle volume:

\[ v = \begin{cases} 
  v_s + v_p & : v_l < 0.85v_p \\
  v_s + 0.15v_p + v_l & : v_l > 0.85v_p.
\end{cases} \tag{9.8} \]

### 9.2.2 Model Extension

It has been observed (Iveson and Litster, 1998; Liu and Litster, 2002) that coagulation of two aggregate particles can occur at two extremes: coagulation of soft particles, resulting in rearrangement of the structure; and coagulation of hard particles, which stick together with no rearrangement. Typically, this has been incorporated into models by considering the entire population of particles to be in one of these extremes at one time (Iveson et al., 2001b). However, the different growth mechanisms depend on the particles’ internal properties (size, wetness etc.) as well as the external properties (e.g. velocity) (Forrest, 2003). It is therefore more realistic to characterise agglomeration behaviour on an individual particle basis, rather than for the population as a whole. Forrest (2003) proposes an extended version of the coagulation probabilities; this is shown in Table 9.2.

Table 9.2: Probability of coalescence as function of properties of granules 1 and 2. The particles are described as either big or small (b/s), dry or wet (d/w) and soft or hard (s/h).

<table>
<thead>
<tr>
<th>1 ↓</th>
<th>b/d/h</th>
<th>b/d/s</th>
<th>b/w/h</th>
<th>b/w/s</th>
<th>s/d/h</th>
<th>s/d/s</th>
<th>s/w/h</th>
<th>s/w/s</th>
</tr>
</thead>
<tbody>
<tr>
<td>b/d/h</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>b/d/s</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>b/w/h</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>b/w/s</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>s/d/h</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>s/d/s</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>s/w/h</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>s/w/s</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
To incorporate these observations and extensions into the model, we wish to consider some measure of the “softness” of the particle. As a simple first approximation, we use the porosity: the higher the porosity of the particle, the more amenable to structural rearrangement it is, i.e., it is softer. In common with the simple, binary, definitions of size and wetness, we characterise the aggregate as either hard or soft according to a critical porosity, \( \varepsilon_{\text{crit}} \):

\[
\begin{align*}
\varepsilon < \varepsilon_{\text{crit}} & : \text{“hard”} \\
\varepsilon > \varepsilon_{\text{crit}} & : \text{“soft”}
\end{align*}
\]

We now have to propose conservation laws for coagulation of aggregates at each of these extreme types of coagulation. Clearly the total solid volume and the total liquid volume must be conserved on coagulation, but what about the total pore volume? It is feasible that on coagulation of two “soft” particles the rearrangement is such that the total pore volume is conserved. But in the case of coagulation of two “hard” particles the total pore volume would increase, due to extra space being “trapped” between the particles. In this case, making the approximation that the aggregates stick together at points only, we might expect that the total surface area of the two particles would be conserved. Note that the surface area being considered here is a notional surface area of all the solid particles making up the aggregate (ignoring any liquid content).

### 9.2.3 Relating Surface Area to Pore Volume

We now have a fourth dimension to characterise the aggregate: the surface area, \( a \). We therefore need to either determine conservation laws for the fourth independent dimension, or to reduce the problem back to three dimensions by defining the surface area in terms of the existing dimensions.

Consider an irregularly shaped aggregate with solid volume \( v_s \), pore volume \( v_p \) and surface area (including the surface area of any internal pores) \( a \) (Figure 9.1). We characterise the aggregate using these variables because we can initially consider \( v_s \) and either \( v_p \) or \( a \) to be conserved on coagulation. The irregularity of the aggregate can be characterised by a surface fractal dimension, such that for
its solid volume, \( v_s \):

\[
\frac{v_s}{v_{s,0}} = \left( \frac{a}{a_0} \right)^{D_S}
\]  

(9.9)

Here, \( v_{s,0} \) and \( a_0 \) are the volume and surface area of a theoretical spherical primary particle of radius \( r_0 \). \( v_{s,0} = \frac{4}{3}\pi r_0^3 \) and \( a_0 = 4\pi r_0^2 \). The surface fractal dimension, \( D_S \), varies from 3/2 (for a perfect sphere) to 1 (for a string of primary particles).

If we can characterise this aggregate with a radius that gives us an estimate of the total enclosed volume (including pore spaces), \( v \), then we will have a good idea of the pore volume of the aggregate.

**Figure 9.1:** Schematic of an aggregate particle. The particle occupies a total volume (including pore spaces) of \( v = \frac{4}{3}\pi r^3 \).

The irregularity of an aggregate can also be characterised by a fractal dimension relating its radius, \( r \), to its solid volume:

\[
\frac{v_s}{v_{k,0}} = \left( \frac{r}{r_0} \right)^{D_F}
\]  

(9.10)

Here, \( D_F \) is 3 for a perfect sphere, and less for non-spherical particles. If we consider the total volume occupied by the aggregate to be \( v = \frac{4}{3}\pi r^3 \), and the
total solid volume to be \( v_s \), then the pore volume is given by:

\[
 v_p = \frac{4}{3} \pi r^3 - v_s \\
= \frac{4}{3} \pi r_0^3 \left( \frac{v_s}{v_{s,0}} \right)^{3/D_F} - v_s \\
= v_{s,0} \left( \frac{v_s}{v_{s,0}} \right)^{3/D_F} - v_s \\
= v_{s,0} \left( \frac{a}{a_0} \right)^{3D_S/D_F} - v_s
\]  

(9.11)

Now depending on the relationship we choose between \( D_S \) and \( D_F \) we have a formula for pore volume, \( v_p \), in terms of solid volume, \( v_s \), and surface area, \( a \). Looking at the simplest case (i.e., a sphere), we have \( D_S = 3/2 \) and \( D_F = 3 \). As agglomerating particles remain roughly spherical, we choose to maintain this relationship and set \( D_S/D_F = 1/2 \). We then find:

\[
 v_p = v_{s,0} \left( \frac{a}{a_0} \right)^{3/2} - v_s \\
= \frac{1}{6 \sqrt{\pi}} a^{3/2} - v_s
\]  

(9.12)

The problem has now been reduced from four dimensions (solid volume, liquid volume, pore volume, surface area) to three.

### 9.2.4 Collision Softness

We now have two extreme types of coagulation, and we can write conservation laws for them both. For a hard (area conserving) collision:

\[
\begin{bmatrix}
 v_{s,1} \\
 v_{l,1} \\
 v_{p,1} \\
 a_1
\end{bmatrix}
+ \begin{bmatrix}
 v_{s,2} \\
 v_{l,2} \\
 v_{p,2} \\
 a_2
\end{bmatrix}
\rightarrow \begin{bmatrix}
 v_{s,1} + v_{s,2} \\
 v_{l,1} + v_{l,2} \\
 v_{p,1} + \frac{1}{\sqrt{\pi}} \frac{(a_1 + a_2)^{3/2} - (v_{s,1} + v_{s,2})}{a_1 + a_2}
\end{bmatrix}
\]  

(9.13)
And for a soft (pore volume conserving) collision:

\[
\begin{bmatrix}
v_{s,1} \\
v_{l,1} \\
v_{p,1} \\
a_1
\end{bmatrix}
+ \begin{bmatrix}
v_{s,2} \\
v_{l,2} \\
v_{p,2} \\
a_2
\end{bmatrix}
\rightarrow \begin{bmatrix}
v_{s,1} + v_{s,2} \\
v_{l,1} + v_{l,2} \\
v_{p,1} + v_{p,2} \\
\left(\frac{a_1}{2} + \frac{a_2}{2}\right)^{2/3}
\end{bmatrix}
\] (9.14)

But what about a collision between a soft particle and a hard one? We can define a collision softness, \(\xi\):

\[
\xi = \frac{a_1(\varepsilon_1 - \varepsilon_{\text{min}}) + a_2(\varepsilon_2 - \varepsilon_{\text{min}})}{(\varepsilon_{\text{crit}} - \varepsilon_{\text{min}})(a_1 + a_2)} \tag{9.15}
\]

with \(\varepsilon\) replaced by \(\varepsilon_{\text{min}}\) if \(\varepsilon < \varepsilon_{\text{min}}\) and \(\varepsilon\) replaced by \(\varepsilon_{\text{crit}}\) if \(\varepsilon > \varepsilon_{\text{crit}}\).

In this case the conservation laws are:

\[
\begin{bmatrix}
v_{s,1} \\
v_{l,1} \\
v_{p,1} \\
a_1
\end{bmatrix}
+ \begin{bmatrix}
v_{s,2} \\
v_{l,2} \\
v_{p,2} \\
a_2
\end{bmatrix}
\rightarrow \begin{bmatrix}
v_{s,1} + v_{s,2} \\
v_{l,1} + v_{l,2} \\
\frac{1}{\sqrt{\pi}} \left[\left(a_1 + a_2\right) + a_1 \left(\frac{a_1}{2} + \frac{a_2}{2}\right)^{2/3} - a_2\right]^{3/2} - \left(\frac{a_1}{2} + \frac{a_2}{2}\right)^{2/3} - \left(v_{s,1} + v_{s,2}\right)
\end{bmatrix}
\] (9.16)

9.2.5 Model Implementation

We now implement our modified model to illustrate the typical results that can be found.

Laurent et al. (2004) found that particle displacement (and therefore velocity) depends on the number of blade passes; therefore we set both collision and compaction to occur at a rate proportional to the rotational speed, \(\omega\).

Forrest (2003) found similar size distributions after the same number of blade revolutions, but different physical properties (such as bulk density or particle strength) as well as observably different structure. Qualitative
reproduction of these observations can be seen in Figure 9.2. For a given number of revolutions, the particle size distributions are similar (Figure 9.2(a)), but the evolution of the average particle porosity is different (Figure 9.2(b)), suggesting different bulk properties could be found.

This model is, of course, still very primitive. No attempt has been made to quantitatively fit simulation results to experimental results, as the experimental results are two few, and the numerical parameters too numerous, to make this worthwhile. Instead, the model should be used in conjunction with carefully designed experiments to verify numerical values and improve phenomenological descriptions. A potential first improvement could be to find a better description of a particle’s softness. Iveson et al. (2001b) defines a Stokes deformation number, $St_{def}$:

$$St_{def} = \frac{\rho u^2}{2\sigma_y},$$

(9.17)

which includes granule properties (the density, $\rho$, and yield stress, $\sigma_y$) and flow properties (the particle velocity, $u$). Getting characteristic particle velocities from (e.g.) positron emission particle tracking (Forrest et al., 2003; Bridgwater et al., 2004), and finding a relationship between yield strength and the particle properties described above (solid volume, liquid volume, pore volume) would enable a more realistic representation of a particle’s softness.
(a) After the same number of blade revolutions, similar size distributions are observed.

(b) Different average porosity suggests that different particle properties can be obtained.

**Figure 9.2:** Illustration of simulated aggregate properties for different values of rotational speed (corresponding to different values of $\varepsilon_{\text{crit}}$). Solid lines show $\varepsilon_{\text{crit}} = 0.1$; dashed lines show $\varepsilon_{\text{crit}} = 0.15$. 
Chapter 10
Conclusions

In this thesis, we have developed and investigated a number of stochastic methods for the solution of population balance problems. The ultimate aim is to have a set of techniques to efficiently solve the multi-dimensional case of the population balance equation.

The starting point for this work was the (one-dimensional) Smoluchowski coagulation equation. Existing stochastic solution methods have proved to be inefficient at solving the Smoluchowski equation when compared with existing deterministic methods (e.g. sectional methods, discretisation, method of moments etc.). Therefore, we applied the Direct Simulation Algorithm with fictitious jumps and developed a new scheme for choosing majorant kernels, based on ensuring that the majorant remains homogeneous, with the same degree as the original coagulation kernel.

To extend the stochastic solution method, we introduced an algorithm for simulating sources and sinks of particles, both when a source of particles is taken from a continuous size distribution and when it is taken from a population of stochastic particles. Comparison of the modified stochastic simulation scheme, with fictitious coagulation events and a source term, showed great improvements in efficiency over the existing stochastic methods.

Application of this method to a physical system, that of synthesis of titania nanoparticles in flames, showed that it can resolve particle size distributions in situations where multiple processes are occurring. At later times, when coagulation is the only process occurring, the system tends to a log-normal distribution (as previously noted). Stochastic modelling has given further insight into the
development of this size distribution and how it might be predicted.

We then included the process of particle breakage in our simulation. As well as the Direct Simulation Algorithm already described, we extended the Mass Flow Algorithm (where one stochastic particle represents a unit of mass concentration rather than number concentration) to the coagulation-breakage problem in the continuous case.

To ensure that the procedure remains as efficient as possible, we investigated several types of data storage/rearrangement. To improve efficiency of selection of particles to undergo coagulation or breakage, we used a binary tree representation to store the required properties of the particles. Selection of the particles (from a population of \( n \)) then takes \( O(\log n) \), as opposed to \( O(n \log n) \) when sorting the particles into bins. Application of a constant number method of simulation (\( i.e. \), with duplication of a stochastic particle if one is removed, or removal of one if an extra particle is added) ensures that the binary tree method of index generation remains simple and efficient to implement.

We simulated a simple test case of coagulation and breakage to compare the performances of DSA and MFA. We found that MFA is much more efficient at simulating any moments of the size distribution higher than the first. However, in a simulation of a physical system, some information about the number concentration or the size distribution is sought. For any property of the size distribution related to the zeroth moment (\( i.e. \), the number of particles in the system), the MFA shows wide fluctuations in its predictions, which can only be resolved by an impractical increase in the number of simulated particles. Therefore, for a problem including breakage (or with no clearly defined lower limit on particle size) it is recommended that the Direct Simulation Algorithm is used.

Application of this numerical method to a physical situation (droplet break-up and coalescence in a rotating disc contactor) allowed some insight into the validity of the physical models that had been assumed. It was found that at low rotational speeds, the phenomenological models, which were based on single droplet behaviour, showed a good prediction of the behaviour of droplet swarms. However, the results of experiment and simulation differed at higher rotational speeds, suggesting qualitatively different behaviour under a different flow regime (a suggestion that has been backed up independently by researchers using computational fluid dynamics simulation).
Extension of the stochastic solution method to the multi-dimensional case was shown to be straightforward, both in implementation and efficiency. A two-dimensional (size and composition) problem was investigated. With the introduction of mass transfer, a continuous process (as opposed to the discrete, identity changing processes investigated previously), several different methods were investigated. Updating every particle after each discrete process time step proved to give an exact solution but was very inefficient. Treating the continuous process as a series of discrete jumps is more efficient but results in a (normally distributed) broadening of the distribution of compositions. The Linear Process Deferment Algorithm gives a sufficiently accurate solution at low computational cost.

Validation of the multi-dimensional stochastic model was done by comparison with an existing numerical solution method (bivariate quadrature method of moments). Despite a slightly different modelling approach, good agreement between the methods was obtained.

Finally, a further application of the multi-dimensional stochastic solution method was discussed. A modified approach to the modelling of wet powder agglomeration was proposed to illustrate how these techniques, in conjunction with experimental insight, can be used to develop predictive models of particle population behaviour.

Overall, stochastic solution methods show many advantages for the solution of population balance problems. They are intuitive to implement, being built around a fictional representation of the real particle population, with each change to the population being an implementation of a real process. It is possible to implement both discrete and continuous processes. A number of techniques have been developed for efficient data storage and retrieval to ensure that stochastic solution methods are an efficient alternative to established deterministic approaches, especially in the multi-dimensional case.
Bibliography


154


BIBLIOGRAPHY


BIBLIOGRAPHY


