## Contents

**Preface**

1 Introduction  
1.1 The Discrete Method ............................................. 1-1  
1.2 The Standard Method of Moments .................................. 1-2  
1.3 The Quadrature Method of Moments ............................... 1-2

2 Population Balance Model Theory  
2.1 The Particle State Vector ........................................ 2-1  
2.2 The Population Balance Equation (PBE) ............................. 2-2  
2.2.1 Particle Growth and Dissolution .................................. 2-2  
2.2.2 Particle Birth and Death Due to Breakage and Aggregation ... 2-3  
2.2.3 Particle Birth by Nucleation .................................... 2-13  
2.3 Solution Methods .................................................. 2-13  
2.3.1 The Discrete Method ........................................... 2-13  
2.3.2 The Standard Method of Moments (SMM) .......................... 2-17  
2.3.3 The Quadrature Method of Moments (QMOM) ..................... 2-19  
2.4 Reconstructing the Particle Size Distribution from Moments .... 2-20

3 Using the ANSYS FLUENT Population Balance Model  
3.1 Population Balance Module Installation ............................ 3-1  
3.2 Loading the Population Balance Module ............................ 3-1  
3.3 Population Balance Model Setup .................................... 3-2  
3.3.1 Enabling the Population Balance Model .......................... 3-2  
3.3.2 Defining Population Balance Boundary Conditions .............. 3-9  
3.3.3 Specifying Population Balance Solution Controls ............... 3-10
4 Postprocessing for the Population Balance Model 4-1
   4.1 Population Balance Solution Variables 4-1
   4.2 Reporting Derived Population Balance Variables 4-2
     4.2.1 Computing Moments 4-2
     4.2.2 Displaying a Number Density Function 4-3

5 UDFs for Population Balance Modeling 5-1
   5.1 Population Balance Variables 5-1
   5.2 Population Balance DEFINE Macros 5-1
     5.2.1 DEFINE_PB_BREAKUP_RATE_FREQ 5-2
     5.2.2 DEFINE_PB_BREAKUP_RATE_PDF 5-3
     5.2.3 DEFINE_PB_COALESCENCE_RATE 5-4
     5.2.4 DEFINE_PB_NUCLEATION_RATE 5-6
     5.2.5 DEFINE_PB_GROWTH_RATE 5-8
   5.3 Hooking a Population Balance UDF to ANSYS FLUENT 5-10

A DEFINE_HET_RXN_RATE Macro A-1
   A.1 Description A-1
   A.2 Usage A-2
   A.3 Example A-3
   A.4 Hooking a Heterogeneous Reaction Rate UDF to ANSYS FLUENT A-4
Using This Manual

The Contents of This Manual

The ANSYS FLUENT Population Balance Model Manual tells you what you need to know to model population balance with ANSYS FLUENT. In this manual, you will find background information pertaining to the model, a theoretical discussion of the model used in ANSYS FLUENT, and a description of using the model for your CFD simulations.

Typographical Conventions

Several typographical conventions are used in this manual’s text to facilitate your learning process.

- An informational icon (i) marks an important note.

- A warning icon (⚠️) marks a warning.

- Different type styles are used to indicate graphical user interface menu items and text interface menu items (e.g., iso-Surface dialog box, surface/iso-surface command).

- The text interface type style is also used when illustrating exactly what appears on the screen or exactly what you need to type into a field in a dialog box. The information displayed on the screen is enclosed in a large box to distinguish it from the narrative text, and user inputs are often enclosed in smaller boxes.

- A mini flow chart is used to guide you through the navigation pane, which leads you to a specific task page or dialog box. For example,

  ![Models ➔ Edit...](image)

  indicates that Models is selected in the navigation pane, which then opens the corresponding task page. In the Models task page, Multi-phase is selected from the list. Clicking the Edit... button opens the Multi-phase dialog box.
Also, a mini flow chart is used to indicate the menu selections that lead you to a specific command or dialog box. For example,

\[
\text{Define } \rightarrow \text{Injections...}
\]

indicates that the Injections... menu item can be selected from the Define pull-down menu, and

\[
\text{display } \rightarrow \text{mesh}
\]

indicates that the mesh command is available in the display text menu.

In this manual, mini flow charts usually precede a description of a dialog box or command, or a screen illustration showing how to use the dialog box or command. They allow you to look up information about a command or dialog box and quickly determine how to access it without having to search the preceding material.

- The menu selections that will lead you to a particular dialog box or task page are also indicated (usually within a paragraph) using a “/”. For example, Define/Materials... tells you to choose the Materials... menu item from the Define pull-down menu.

Mathematical Conventions

- Where possible, vector quantities are displayed with a raised arrow (e.g., \( \vec{a}, \vec{A} \)). Boldfaced characters are reserved for vectors and matrices as they apply to linear algebra (e.g., the identity matrix, \( \mathbf{I} \)).

- The operator \( \nabla \), referred to as grad, nabla, or del, represents the partial derivative of a quantity with respect to all directions in the chosen coordinate system. In Cartesian coordinates, \( \nabla \) is defined to be

\[
\frac{\partial}{\partial x} \vec{i} + \frac{\partial}{\partial y} \vec{j} + \frac{\partial}{\partial z} \vec{k}
\]

\( \nabla \) appears in several ways:

- The gradient of a scalar quantity is the vector whose components are the partial derivatives; for example,

\[
\nabla p = \frac{\partial p}{\partial x} \vec{i} + \frac{\partial p}{\partial y} \vec{j} + \frac{\partial p}{\partial z} \vec{k}
\]
– The gradient of a vector quantity is a second-order tensor; for example, in Cartesian coordinates,

\[ \nabla(\vec{v}) = \left( \frac{\partial}{\partial x} \vec{\hat{i}} + \frac{\partial}{\partial y} \vec{\hat{j}} + \frac{\partial}{\partial z} \vec{\hat{k}} \right) \left( v_x \vec{\hat{i}} + v_y \vec{\hat{j}} + v_z \vec{\hat{k}} \right) \]

This tensor is usually written as

\[
\begin{pmatrix}
\frac{\partial v_x}{\partial x} & \frac{\partial v_x}{\partial y} & \frac{\partial v_x}{\partial z} \\
\frac{\partial v_y}{\partial x} & \frac{\partial v_y}{\partial y} & \frac{\partial v_y}{\partial z} \\
\frac{\partial v_z}{\partial x} & \frac{\partial v_z}{\partial y} & \frac{\partial v_z}{\partial z}
\end{pmatrix}
\]

– The divergence of a vector quantity, which is the inner product between \( \nabla \) and a vector; for example,

\[ \nabla \cdot \vec{v} = \frac{\partial v_x}{\partial x} + \frac{\partial v_y}{\partial y} + \frac{\partial v_z}{\partial z} \]

– The operator \( \nabla \cdot \nabla \), which is usually written as \( \nabla^2 \) and is known as the Laplacian; for example,

\[ \nabla^2 T = \frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} + \frac{\partial^2 T}{\partial z^2} \]

\( \nabla^2 T \) is different from the expression \((\nabla T)^2\), which is defined as

\[ (\nabla T)^2 = \left( \frac{\partial T}{\partial x} \right)^2 + \left( \frac{\partial T}{\partial y} \right)^2 + \left( \frac{\partial T}{\partial z} \right)^2 \]
Technical Support

If you encounter difficulties while using ANSYS FLUENT, please first refer to the section(s) of the manual containing information on the commands you are trying to use or the type of problem you are trying to solve. The product documentation is available from the online help, or from the User Services Center (www.fluentusers.com).

If you encounter an error, please write down the exact error message that appeared and note as much information as you can about what you were doing in ANSYS FLUENT. Then refer to the following resources available on the User Services Center (www.fluentusers.com):

- Installation and System FAQs - link available from the main page on the User Services Center. The FAQs can be searched by word or phrase, and are available for general installation questions as well as for products.

- Known defects for ANSYS FLUENT - link available from the product page. The defects can be searched by word or phrase, and are listed by categories.

- Online Technical Support - link available from the main page on the User Services Center. From the Online Technical Support Portal page, there is a link to the Search Solutions & Request Support page, where the solutions can be searched by word or phrase.

Contacting Technical Support

If none of the resources available on the User Services Center help in resolving the problem, or you have complex modeling projects, we invite you to log a technical support request (www.fluentusers.com) to obtain further assistance. However, there are a few things that we encourage you to do before logging a request:

- Note what you are trying to accomplish with ANSYS FLUENT.

- Note what you were doing when the problem or error occurred.

- Save a journal or transcript file of the ANSYS FLUENT session in which the problem occurred. This is the best source that we can use to reproduce the problem and thereby help to identify the cause.
Chapter 1. Introduction

In ANSYS FLUENT the population balance model is provided as an addon module with the standard ANSYS FLUENT licensed software.

Several industrial fluid flow applications involve a secondary phase with a size distribution. The size distribution of particles, including solid particles, bubbles, or droplets, can evolve in conjunction with transport and chemical reaction in a multiphase system. The evolutionary processes can be a combination of different phenomena like nucleation, growth, dispersion, dissolution, aggregation, and breakage producing the dispersion. Thus in multiphase flows involving a size distribution, a balance equation is required to describe the changes in the particle population, in addition to momentum, mass, and energy balances. This balance is generally referred to as the population balance. Cases in which a population balance could apply include crystallization, precipitative reactions from a gas or liquid phase, bubble columns, gas sparging, sprays, fluidized bed polymerization, granulation, liquid-liquid emulsion and separation, and aerosol flows.

To make use of this modeling concept, a number density function is introduced to account for the particle population. With the aid of particle properties (e.g., particle size, porosity, composition, etc.), different particles in the population can be distinguished and their behavior can be described.

ANSYS FLUENT offers three solution methods to the population balance equation: discretized population balance, standard method of moments, and quadrature method of moments.

- Section 1.1: The Discrete Method
- Section 1.2: The Standard Method of Moments
- Section 1.3: The Quadrature Method of Moments

1.1 The Discrete Method

In the discrete method, the particle population is discretized into a finite number of size intervals. This approach has the advantage of computing the particle size distribution (PSD) directly. This approach is also particularly useful when the range of particle sizes is known a priori and does not span more than two or three orders of magnitude. In this case, the population can be discretized with a relatively small number of size intervals and the size distribution that is coupled with fluid dynamics can be computed. The disadvantage of the discrete method is that it is computationally expensive if a large number of intervals is needed.
1.2 The Standard Method of Moments

The standard method of moments (SMM) is an efficient alternative to the discrete population balance approach. In this approach, the population balance equation is transformed into a set of transport equations for moments of the distribution. The $i$th moment is defined by integrating the number density throughout the particle space weighted with the particle property raised to its $i$th power. It is generally sufficient to solve only a few moment equations, typically three to six. This may provide a significant reduction in the number of equations to be solved compared with the discretized approach. Apart from the computational advantage, the SMM approach is useful when the entire distribution is not needed and certain average and total quantities are sufficient to represent the particle distribution. Typically, the zeroth moment represents the total number density, the second moment represents the total surface area per unit volume, and the third moment represents the total mass density.

In the SMM approach, no assumptions are made about the size distribution, and the moment equations are formulated in a closed form involving only functions of the moments themselves. However, this exact closure requirement poses a serious limitation, as aggregation (with the exception of the constant aggregation kernel) and breakage phenomena cannot be written as functions of moments.

1.3 The Quadrature Method of Moments

The quadrature method of moments (QMOM) has a similar advantage as the SMM in terms of computational costs, but replaces the exact closure needed by SMM with an approximate closure. This allows application of QMOM to a broad range of applications without any limitations.
Chapter 2. Population Balance Model Theory

This chapter presents an overview of the theory and the governing equations for the methods used in ANSYS FLUENT to predict particle growth and nucleation.

- Section 2.1: The Particle State Vector
- Section 2.2: The Population Balance Equation (PBE)
- Section 2.3: Solution Methods
- Section 2.4: Reconstructing the Particle Size Distribution from Moments

2.1 The Particle State Vector

The particle state vector is characterized by a set of “external coordinates” \((\vec{x})\), which denote the spatial position of the particle, and “internal coordinates” \((\phi)\), which could include particle size, composition, and temperature. From these coordinates, a number density function \(n(\vec{x}, \phi, t)\) can be postulated where \(\phi \in \Omega_\phi, \vec{x} \in \Omega_{\vec{x}}\). Therefore, the average number of particles in the infinitesimal volume \(d\vec{x}d\phi\) is \(n(\vec{x}, \phi, t)d\vec{x}d\phi\). In contrast, the continuous phase state vector is given by \(\vec{Y} \equiv [Y_1(\vec{x}, t), Y_2(\vec{x}, t), \ldots, Y_c(\vec{x}, t)]\)

The total number of particles in the entire system is then

\[
\int_{\Omega_\phi} \int_{\Omega_{\vec{x}}} n \, d\vec{x}d\phi
\]

The local average number density in physical space (i.e., the total number of particles per unit volume) is given by

\[
N(\vec{x}, t) = \int_{\Omega_{\vec{x}}} n \, d\phi
\]

The total volume fraction of all particles is given by

\[
\alpha(\vec{x}, t) = \int_{\Omega_\phi} V(\phi) d\phi
\]

where \(V(\phi)\) is the volume of a particle in state \(\phi\).
2.2 The Population Balance Equation (PBE)

Assuming that $\phi$ is the particle volume, the transport equation for the number density function is given as

$$\frac{\partial}{\partial t} [n(V, t)] + \nabla \cdot [\vec{u} n(V, t)] + \nabla_v \cdot [G_v n(V, t)] =\]

- **Growth term**

\[
\frac{1}{2} \int_0^V a(V - V', V') n(V - V', t)n(V', t)dV' - \int_0^\infty a(V, V') n(V, t)n(V', t)dV'
\]

- **Birth due to Aggregation**

- **Death due to Aggregation**

\[
+ \int_{\Omega_v} p g(V') \beta(V' | V') n(V', t)dV' - g(V)n(V, t)
\]

- **Birth due to Breakage**

- **Death due to Breakage**

The boundary and initial conditions are given by

$$n(V, t = 0) = n_v; \quad n(V = 0, t)G_v = \dot{n}_0$$

(2.2-2)

where $\dot{n}_0$ is the nucleation rate in particles/m$^3$-s.

2.2.1 Particle Growth and Dissolution

The growth rate based on particle volume, $G_v$, (m$^3$/s) is defined as

$$G_v = \frac{\partial V}{\partial t}$$

(2.2-3)

The growth rate based on particle diameter (or length) is defined as

$$G = \frac{\partial L}{\partial t}$$

(2.2-4)

The volume of a single particle $V$ is defined as $K_v L^3$, and therefore the relationship between $G_v$ and $G$ is

$$G_v = 3K_v L^2 G$$

(2.2-5)

The surface area of a single particle, $A$, is defined as $K_a L^2$. Thus for a cube or a sphere, $K_a = 6K_v$.

Dissolution of particles can be represented as negative growth.
2.2 The Population Balance Equation (PBE)

2.2.2 Particle Birth and Death Due to Breakage and Aggregation

The birth and death of particles occur due to breakage and aggregation processes. Examples of breakage processes include crystal fracture in crystallizers and bubble breakage due to liquid turbulence in a bubble column. Similarly, aggregation can occur due to particle agglomeration in crystallizers and bubble coalescence in bubble column reactors.

**Breakage**

The breakage rate expression, or kernel [18], is expressed as

\[ g(V') \beta(V | V') \]

where

- \( g(V') \) = breakage frequency; i.e., the fraction of particles of volume \( V' \) breaking per unit time \( (m^{-3}s^{-1}) \)
- \( \beta(V | V') \) = probability density function (PDF) of particles breaking from volume \( V' \) to a particle of volume \( V \)

The birth rate of particles of volume \( V \) due to breakage is given by

\[ B_{br} = \int_{\Omega_V} pg(V')\beta(V | V')n(V')dV' \quad (2.2-6) \]

where \( pg(V')n(V')dV' \) particles of volume \( V' \) break per unit time, producing \( pg(V')n(V')dV' \) particles, of which a fraction \( \beta(V | V')dV \) represents particles of volume \( V \). \( p \) is the number of child particles produced per parent (e.g., two particles for binary breakage).

The death rate of particles of volume \( V \) due to breakage is given by

\[ D_{br} = g(V)n(V) \quad (2.2-7) \]

The PDF \( \beta(V | V') \) is also known as the particle fragmentation distribution function, or daughter size distribution. Several functional forms of the fragmentation distribution function have been proposed, though the following physical constraints must be met: the normalized number of breaking particles must sum to unity, the masses of the fragments must sum to the original particle mass, and the number of fragments formed has to be correctly represented.
Mathematically, these constraints can be written as follows:

- For the normalization condition:
  \[ \int_{0}^{V'} \beta(V | V') dV = 1 \]  \hspace{1cm} (2.2-8)

- For conservation of mass
  \[ p \int_{0}^{V'} m(V) \beta(V | V') dV = m(V') \]  \hspace{1cm} (2.2-9)

- For binary breakage, \( \beta \) is symmetric about \( V/V' = 0.5 \); i.e.,
  \[ \beta(V' - V | V') = \beta(V | V') \]  \hspace{1cm} (2.2-10)

The following is a list of models available in ANSYS FLUENT to calculate the breakage frequency:

- constant value
- Luo model
- Lehr model
- Ghadiri model
- user-defined model

ANSYS FLUENT provides the following models for calculating the breakage PDF:

- parabolic PDF
- generalized PDF for multiple breakage fragments
- user-defined model

The breakage frequency models and the parabolic and generalized PDFs are described in detail in the sections that follow.
Luo and Lehr Breakage Kernels

The Luo and Lehr models are integrated kernels, encompassing both the breakage frequency and the PDF of breaking particles. The general breakage rate per unit volume is usually written as

$$\Omega_{br} (V, V') = \Omega_B (V') \eta (V \mid V') [1/m^3/sec] \quad (2.2-11)$$

where the original particle has a volume $V'$ and the daughter particle has a volume $V$. In the previous expression, $\Omega_B (V')$ is the breakage frequency, and $\eta (V \mid V')$ is the normalized daughter particle distribution function. For binary breakage, the breakage kernel must be symmetrical with respect to $\frac{V}{V'} = 0.5$.

The general form is the integral over the size of eddies $\lambda$ hitting the particle with diameter $d$ (and volume $V$). The integral is taken over the dimensionless eddy size $\xi = \lambda/d$. The general form is

$$\Omega_{br} (V, V') = K \int_{\xi_{min}}^{1} \frac{(1 + \xi)^2}{\xi^n} \exp (-b \xi^m) \, d\xi \quad (2.2-12)$$

where the parameters are as shown in Table 2.2.1:

Table 2.2.1: Luo and Lehr Model Parameters

<table>
<thead>
<tr>
<th></th>
<th>$K [1/m^3/sec]$</th>
<th>$n$</th>
<th>$b$</th>
<th>$m$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Luo</td>
<td>$0.9238 \varepsilon^{1/3} d^{-2/3} \alpha$</td>
<td>11/3</td>
<td>$12 \left( f^{2/3} + (1 - f)^{2/3} - 1 \right) \sigma \rho^{-1} \varepsilon^{-2/3} d^{-5/3}$</td>
<td>$-11/3$</td>
</tr>
<tr>
<td>Lehr</td>
<td>$1.19 \varepsilon^{-1/3} d^{-1/3} \sigma \rho^{-1} f^{-1/3}$</td>
<td>13/3</td>
<td>$2 \sigma \rho^{-1} \varepsilon^{-2/3} d^{-5/3} f^{-1/3}$</td>
<td>$-2/3$</td>
</tr>
</tbody>
</table>
Ghadiri Breakage Kernels

The Ghadiri model [7, 22], in contrast to the Luo and Lehr models, is used to model only the breakage frequency of the solid particles. You will have to specify the PDF model to define the daughter distribution.

The breakage frequency $f$ is related to the material properties and impact conditions:

$$f = \frac{\rho_s E^{2/3}}{\Gamma^{5/3}} v^2 L^{5/3} = K_b v^2 L^{5/3} \quad (2.2-13)$$

where $\rho_s$ is the particle density, $E$ is the elastic modulus of the granule, and $\Gamma$ is the interface energy. $v$ is the impact velocity and $L$ is the particle diameter prior to breaking. $K_b$ is the breakage constant and is defined as

$$K_b = \frac{\rho_s E^{2/3}}{\Gamma^{5/3}} \quad (2.2-14)$$

Parabolic PDF

The breakage PDF function contains information on the probability of fragments formed by a breakage event. It provides the number of particles and the possible size distribution from the breakage. The parabolic form of the PDF implemented in ANSYS FLUENT allows you to define the breakage PDF such that

$$\beta (V | V') = 0.5 \left[ \frac{C}{V'} + \frac{1 - C/2}{V} \left\{ 24 \left( \frac{V}{V'} \right)^2 - 24 \left( \frac{V}{V'} \right) + 6 \right\} \right] \quad (2.2-15)$$

where $V$ and $V'$ are the daughter and parent particle volumes, respectively. Depending on the value of the shape factor $C$, different behaviors will be observed in the shape of the particle breakage distribution function. For example, if $C = 2$, the particle breakage has a uniform distribution. If $0 < C < 2$, a concave parabola is obtained, meaning that it is more likely to obtain unequally-sized fragments than equally-sized fragments. The opposite of this is true for $2 < C < 3$. Values outside of the range of 0 to 3 are not allowed, because the PDF cannot have a negative value.

Note that the PDF defined in Equation 2.2-15 is symmetric about $V/V' = 0.5$. 
Generalized PDF

The generalized form of the PDF implemented in ANSYS FLUENT allows you to simulate multiple breakage fragments (> 2) and to specify the form of the daughter distribution (e.g., uniform, equisized, attrition, power law, parabolic, binary beta). The model itself can be applied to both the discrete method and the QMOM.

Considering the self-similar formulation [25] where the similarity z is the ratio of daughter-to-parent size (i.e., \( z \equiv \frac{V}{V'} \)), then the generalized PDF is given by

\[
p\beta(V|V') = \frac{\theta(z)}{V'}
\]  

(2.2-16)

The \( k^{th} \) moment of \( \theta(z) \) (\( b_k \)) is

\[
b_k = \int_0^1 z^k \theta(z) \, dz = \frac{B_k(V')}{V'^k}
\]  

(2.2-17)

where

\[
B_k(V') = \int_0^{V'} V^k p\beta(V|V') \, dV
\]  

(2.2-18)

The conditions of number and mass conservation can then be expressed as

\[
b_0 = \int_0^1 \theta(z) \, dz = p
\]  

(2.2-19)

\[
b_1 = \int_0^1 z \theta(z) \, dz = 1
\]  

(2.2-20)

The generalized form of \( \theta(z) \)[6] can be expressed as

\[
\theta(z) = \sum_i w_i p_i z^{q_i-1}(1-z)^{r_i-1} \frac{1}{\beta(q_i, r_i)}
\]  

(2.2-21)

where \( i \) can be 0 or 1, which represents \( \theta(z) \) as consisting of 1 or 2 terms, respectively. For each term, \( w_i \) is the weighting factor, \( p_i \) is the averaged number of daughter particles, \( q_i \) and \( r_i \) are the exponents, and \( \beta(q_i, r_i) \) is the beta function. The following constraints are imposed on the parameters in Equation 2.2-21:

\[
\sum_i w_i = 1
\]  

(2.2-22)
\[ \sum_i w_i p_i = p \]  \hspace{1cm} (2.2-23)

\[ \sum_i w_i \left( \frac{p_i q_i}{q_i + r_i} \right) = 1 \]  \hspace{1cm} (2.2-24)

In order to demonstrate how to transform the generalized PDF to represent an appropriate daughter distribution, consider the expressions shown in Table 2.2.2:

<table>
<thead>
<tr>
<th>Type</th>
<th>( \theta(z) )</th>
<th>( p )</th>
<th>Constraints</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equisized[12]</td>
<td>( p \delta(z - \frac{1}{p}) )</td>
<td>( p )</td>
<td>( p \geq 2 )</td>
</tr>
<tr>
<td>Attrition[12]</td>
<td>( \delta(z - (1 - \varepsilon)) + \delta(z - \varepsilon) )</td>
<td>( 2 )</td>
<td>( \varepsilon \ll 2 )</td>
</tr>
<tr>
<td>Power Law[29]</td>
<td>((\nu + 1) z^{\nu - 1} ) \frac{\nu+1}{\nu}</td>
<td>( 0 &lt; \nu \leq 1 )</td>
<td></td>
</tr>
<tr>
<td>Parabolic -a[29]</td>
<td>((\nu + 2)(\nu + 1)z^{\nu - 1}(1 - z) ) \frac{\nu+2}{\nu}</td>
<td>( 0 &lt; \nu \leq 2 )</td>
<td></td>
</tr>
<tr>
<td>Austin[2]</td>
<td>( w(\nu_1 + 1) z^{\nu_1 - 1} ) (+)(1 - w)(\nu_2 + 1) z^{\nu_2 - 1} )</td>
<td>( w(1 + \frac{1}{\nu_1}) ) (+(1 - w)(1 + \frac{1}{\nu_2}) )</td>
<td>( \nu_1, \nu_2 &gt; 0 )</td>
</tr>
<tr>
<td>Binary Beta -a[11]</td>
<td>( 60 z^2 (1 - z)^2 ) (2 )</td>
<td>N/A</td>
<td></td>
</tr>
<tr>
<td>Binary Beta -b[20]</td>
<td>( \frac{z^2}{\beta(\nu, \nu)} z^{\nu - 1}(1 - z)^{\nu-1} ) (2 )</td>
<td>( \nu &gt; 0 )</td>
<td></td>
</tr>
<tr>
<td>Uniform[29]</td>
<td>( p(p - 1)(1 - z)^{p-2} ) (p )</td>
<td>( p \geq 2 )</td>
<td></td>
</tr>
</tbody>
</table>

In Table 2.2.2, \( \delta \) is the Dirac delta function, \( w \) is a weighting coefficient, and \( \varepsilon, \nu, \nu_1, \) and \( \nu_2 \) are user-defined parameters.

The generalized form can represent the daughter distributions in Table 2.2.2 by using the values shown in Table 2.2.3.
Table 2.2.3: Values for Daughter Distributions in General Form

<table>
<thead>
<tr>
<th>Type</th>
<th>$w_0$</th>
<th>$p_0$</th>
<th>$q_0$</th>
<th>$r_0$</th>
<th>$w_1$</th>
<th>$p_1$</th>
<th>$q_1$</th>
<th>$r_1$</th>
<th>Constraints</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equisized</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>$\infty$*</td>
<td></td>
<td>$p \geq 2$</td>
</tr>
<tr>
<td>Attrition</td>
<td>0.5</td>
<td>2</td>
<td>$\varepsilon$</td>
<td>1</td>
<td>0.5</td>
<td>2</td>
<td>1</td>
<td>$\varepsilon$</td>
<td>$\varepsilon \ll 1$</td>
</tr>
<tr>
<td>Power Law</td>
<td>1</td>
<td>$\nu + 1$</td>
<td>$\nu$</td>
<td>1</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>$0 &lt; \nu \leq 1$</td>
</tr>
<tr>
<td>Parabolic</td>
<td>1</td>
<td>$\nu + 2$</td>
<td>$\nu$</td>
<td>2</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>$0 &lt; \nu \leq 2$</td>
</tr>
<tr>
<td>Austin</td>
<td>$w$</td>
<td>$\nu_1 + 1$</td>
<td>$\nu_1$</td>
<td>1</td>
<td>$1 - w$</td>
<td>$\nu_2$</td>
<td>1</td>
<td>$\nu_1, \nu_2 &gt; 0$</td>
<td>$1 \geq w \geq \nu_1 (\frac{\nu_2 - 1}{\nu_2 - \nu_1})$</td>
</tr>
<tr>
<td>Binary Beta**</td>
<td>1</td>
<td>2</td>
<td>$\nu$</td>
<td>$\nu$</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>$\nu &gt; 0$</td>
</tr>
<tr>
<td>Uniform</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>$p \geq 2$</td>
</tr>
</tbody>
</table>

(*) You can approximate $\infty$ by using a very large number, such as $1e10$.
(**) Binary Beta -a is a special case of Binary Beta -b when $\nu = 3$.

Note that for the ANSYS FLUENT implementation of the generalized form of the PDF, you will only enter values for $w_0$, $p_0$, $q_0$, $r_0$, and $q_1$, and the remaining values ($w_1$, $p_1$, and $r_1$) will be calculated automatically.

**Aggregation**

The aggregation kernel [18] is expressed as

$$a(V, V')$$

The aggregation kernel has units of $m^3/s$, and is sometimes defined as a product of two quantities:

- the frequency of collisions between particles of volume $V$ and particles of volume $V'$
- the “efficiency of aggregation” (i.e., the probability of particles of volume $V$ coalescing with particles of volume $V'$).

The birth rate of particles of volume $V$ due to aggregation is given by

$$B_{ag} = \frac{1}{2} \int_0^V a(V - V', V')n(V - V')n(V')dV' \quad (2.2-25)$$

where particles of volume $V - V'$ aggregate with particles of volume $V'$ to form particles of volume $V$. The factor $1/2$ is included to avoid accounting for each collision event twice.
The death rate of particles of volume $V$ due to aggregation is given by

$$D_{ag} = \int_0^\infty a(V, V') n(V) n(V') dV'$$  \tag{2.2-26}$$

The breakage and aggregation kernels depend on the nature of the physical application. For example, in gas-liquid dispersion, the kernels are functions of the local liquid-phase turbulent dissipation.

The following is a list of aggregation functions available in ANSYS FLUENT:

- constant
- Luo model
- Free molecular model
- Turbulent model
- user-defined model

The Luo, free molecular, and turbulent aggregation functions are described in detail in the sections that follow.

**Luo Aggregation Kernel**

For the Luo model \cite{17}, the general aggregation kernel is defined as the rate of particle volume formation as a result of binary collisions of particles with volumes $V_i$ and $V_j$:

$$\Omega_{ag} (V_i, V_j) = \omega_{ag} (V_i, V_j) P_{ag} (V_i, V_j) \text{[m}^3/\text{sec]}$$  \tag{2.2-27}$$

where $\omega_{ag} (V_i, V_j) \text{[m}^3/\text{sec]}$ is the frequency of collision and $P_{ag} (V_i, V_j)$ is the probability that the collision results in coalescence. The frequency is defined as follows:

$$\omega_{ag} (V_i, V_j) = \frac{\pi}{4} \left( d_i^2 + d_j^2 \right) n_i n_j \pi_{ij}$$  \tag{2.2-28}$$

where $\pi_{ij}$ is the characteristic velocity of collision of two particles with diameters $d_i$ and $d_j$ and number densities $n_i$ and $n_j$. Two physical mechanisms are behind the calculation of this velocity. The first mechanism is turbulent mixing. Assuming that the particles’ size lies in the inertial range of turbulence, and the turbulence is isotropic, the mixing velocity $\pi_{ij}^t$ of the two particles can be expressed as

$$\pi_{ij}^t = \left( \pi_i^2 + \pi_j^2 \right)^{1/2}$$  \tag{2.2-29}$$
where

\[ \pi_i = 1.43 (\varepsilon d_i)^{1/3} \]  

The expression for the probability of aggregation is

\[
P_{ag} = \exp \left\{ -c_1 \frac{0.75 \left(1 + x_{ij}^2 \right) \left(1 + x_{ij}^3 \right)^{1/2}}{(\rho_2/\rho_1 + 0.5)^{1/2}(1 + x_{ij}^3)^{1/2}} \text{We}_{ij}^{1/2} \right\}
\]

(2.2-31)

where \( c_1 \) is a constant of order unity, \( x_{ij} = d_i/d_j \), \( \rho_1 \) and \( \rho_2 \) are the densities of the primary and secondary phases, respectively, and the Weber number is defined as

\[
\text{We}_{ij} = \frac{\rho_i d_i (\pi_{ij})^2}{\sigma}
\]

(2.2-32)

**Free Molecular Aggregation Kernel**

Real particles aggregate and break with frequencies (or kernels) characterized by complex dependencies over particle internal coordinates [28]. In particular, very small particles (say up to \( 1 \mu m \)) aggregate because of collisions due to Brownian motions. In this case, the frequency of collision is size-dependent and usually the following kernel is implemented:

\[
a(L_i, L_j) = \frac{2k_B T (L_i + L_j)^2}{3\mu L_i L_j}
\]

(2.2-33)

where \( k_B \) is the Boltzmann constant, \( T \) is the absolute temperature, \( \mu \) is the viscosity of the suspending fluid. This kernel is also known as the Brownian kernel or the perikinetic kernel.

**Turbulent Aggregation Kernel**

During mixing processes, mechanical energy is supplied to the fluid. This energy creates turbulence within the fluid. The turbulence creates eddies, which in turn help dissipate the energy. The energy is transferred from the largest eddies to the smallest eddies in which it is dissipated through viscous interactions. The size of the smallest eddies is the Kolmogorov microscale, \( \eta \), which is expressed as a function of the kinematic viscosity and the turbulent energy dissipation rate:

\[
\eta = \left( \frac{\nu^3}{\epsilon} \right)^{1/4}
\]

(2.2-34)

In the turbulent flow field, aggregation can occur by two mechanisms:
• viscous subrange mechanism: this is applied when particles are smaller than the Kolmogorov microscale, \( \nu \)

• inertial subrange mechanism: this is applied when particles are bigger than the Kolmogorov microscale. In this case, particles assume independent velocities.

For the viscous subrange, particle collisions are influenced by the local shear within the eddy. Based on work by Saffman and Turner [27], the collision rate is expressed as,

\[
a(L_i, L_j) = \varsigma_T \sqrt{\frac{8\pi}{15} \dot{\gamma}(L_i + L_j)^3} \frac{1}{8} \tag{2.2-35}
\]

where \( \varsigma_T \) is a pre-factor that takes into account the capture efficiency coefficient of turbulent collision, and \( \dot{\gamma} \) is the shear rate:

\[
\dot{\gamma} = \frac{\epsilon^{0.5}}{v} \tag{2.2-36}
\]

For the inertial subrange, particles are bigger than the smallest eddy, therefore they are dragged by velocity fluctuations in the flow field. In this case, the aggregation rate is expressed using Abrahamson’s model [1],

\[
a(L_i, L_j) = \varsigma_T 2^{3/2} \sqrt{\frac{(L_i + L_j)^2}{4 \sqrt{(U_i^2 + U_j^2)}}} \tag{2.2-37}
\]

where \( U_i^2 \) is the mean squared velocity for particle \( i \).

The empirical capture efficiency coefficient of turbulent collision describes the hydrodynamic and attractive interaction between colliding particles. Higashitani et al. [9] proposed the following relationship:

\[
\varsigma_T = 0.732 \left( \frac{5}{N_T} \right)^{0.242} ; N_T \geq 5 \tag{2.2-38}
\]

where \( N_T \) is the ratio between the viscous force and the Van der Waals force,

\[
N_T = \frac{6\pi \mu (L_i + L_j)^3 \dot{\lambda}}{8H} \tag{2.2-39}
\]

Where \( H \) is the Hamaker constant, a function of the particle material, and \( \dot{\lambda} \) is the deformation rate,

\[
\dot{\lambda} = \left( \frac{4\epsilon}{15\pi v} \right)^{0.5} \tag{2.2-40}
\]
2.2.3 Particle Birth by Nucleation

Depending on the application, spontaneous nucleation of particles can occur due to the transfer of molecules from the primary phase. For example, in crystallization from solution, the first step is the phase separation or “birth” of new crystals. In boiling applications, the creation of the first vapor bubbles is a nucleation process referred to as nucleate boiling.

The nucleation rate is defined through a boundary condition as shown in Equation 2.2-2.

2.3 Solution Methods

As discussed in Chapter 1: Introduction, the population balance equation can be solved by three different methods in ANSYS FLUENT: the discrete method, the standard method of moments (SMM), and the quadrature method of moments (QMOM). For each method, the ANSYS FLUENT implementation is limited to a single internal coordinate corresponding to particle size. The following subsections describe the theoretical background of each method and list their advantages and disadvantages.

2.3.1 The Discrete Method

The discrete method (also known as the classes or sectional method) was developed by Hounslow [10], Litster [16], and Ramkrishna [25]. It is based on representing the continuous particle size distribution (PSD) in terms of a set of discrete size classes or bins, as illustrated in Figure 2.3.1. The advantages of this method are its robust numerics and that it gives the PSD directly. The disadvantages are that the bins must be defined \textit{a priori} and that a large number of classes may be required.
Numerical Method

In ANSYS FLUENT, the PBE is written in terms of volume fraction of particle size $i$:

$$\frac{\partial}{\partial t}(\rho_s \alpha_i) + \nabla \cdot (\rho_s u_i \alpha_i) + \frac{\partial}{\partial V} \left( \frac{G_v \rho_s \alpha_i}{V} \right) = \rho_s V_i (B_{ag,i} - D_{ag,i} + B_{br,i} - D_{br,i}) + 0^i \rho_s V_0 \dot{n}_0$$

(2.3-1)

where $\rho_s$ is the density of the secondary phase and $\alpha_i$ is the volume fraction of particle size $i$, defined as

$$\alpha_i = N_i V_i \quad i = 0, 1, \ldots, N - 1$$

(2.3-2)

where

$$N_i(t) = \int_{V_i}^{V_{i+1}} n(V,t)dV$$

(2.3-3)

and $V_i$ is the volume of the particle size $i$. In ANSYS FLUENT, a fraction of $\alpha$, called $f_i$, is introduced as the solution variable. This fraction is defined as

$$f_i = \frac{\alpha_i}{\alpha}$$

(2.3-4)

where $\alpha$ is the total volume fraction of the secondary phase.
The nucleation rate \( \dot{n}_0 \) appears in the discretized equation for the volume fraction of the smallest size \( V_0 \). The notation \( ^0i \) signifies that this particular term, in this case \( \rho_s V_0 \dot{n}_0 \), appears in Equation 2.3-1 only in the case of the smallest particle size.

The growth rate in Equation 2.3-1 is discretized as follows [10]:

\[
\frac{\partial}{\partial V} \left( \frac{G_v \rho_s \alpha_i}{V} \right) = \rho_s V_i \left[ \left( \frac{G_{v,i-1} N_{i-1}}{V_i - V_{i-1}} \right) - \left( \frac{G_{v,i} N_i}{V_{i+1} - V_i} \right) \right] \tag{2.3-5}
\]

The volume coordinate is discretized as [10] \( V_{i+1}/V_i = 2^q \) where \( q = 1, 2, \ldots \) and is referred to as the “ratio factor”.

The particle birth and death rates are defined as follows:

\[
B_{ag,i} = \sum_{k=1}^{N} \sum_{j=1}^{N} a_{kj} N_k N_j x_{kj} \xi_{kj} \tag{2.3-6}
\]

\[
D_{ag,i} = \sum_{j=1}^{N} a_{ij} N_i N_j \tag{2.3-7}
\]

\[
B_{br,i} = \sum_{j=i+1}^{N} g(V_j) N_j \beta(V_i | V_j) \tag{2.3-8}
\]

\[
D_{br,i} = g(V_i) N_i \tag{2.3-9}
\]

where \( a_{ij} = a(V_i, V_j) \) and

\[
\xi_{kj} = \begin{cases} 
1 & \text{for } V_i < V_{ag} < V_{i+1}, \text{ where } i \leq N - 1 \\
0 & \text{otherwise}
\end{cases} \tag{2.3-10}
\]

\( V_{ag} \) is the particle volume resulting from the aggregation of particles \( k \) and \( j \), and is defined as

\[
V_{ag} = [x_{kj} V_i + (1 - x_{kj}) V_{i+1}] \tag{2.3-11}
\]

where

\[
x_{kj} = \frac{V_{ag} - V_{i+1}}{V_i - V_{i+1}} \tag{2.3-12}
\]
If \( V_{ag} \) is greater than or equal to the largest particle size \( V_N \), then the contribution to class \( N - 1 \) is

\[
x_{kj} = \frac{V_{ag}}{V_N}
\]

\[ (2.3-13) \]

\[ \text{Note that there is no breakage for the smallest particle class.} \]

**Breakage Formulations for the Discrete Method**

The default breakage formulation for the discrete method in ANSYS FLUENT is based on the Hagesather method [14]. In this method, the breakage sources are distributed to the respective size bins, preserving mass and number density. For the case when the ratio between successive bin sizes can be expressed as \( 2^n \) where \( n = 1, 2, \ldots \), the source in bin \( i \), \( (i = 1, \ldots, N) \) can be expressed as

\[
B_b(i) = \sum_{k=i+1, k \neq N}^{N} \Omega_b(v_k, v_i) + \sum_{k=i, k \neq N}^{i} x_{i+1,k} \Omega_b(v_{i+1}, v_k) + \sum_{k=1, k \neq 1}^{i-1} (1 - x_{i,k}) g(v_{i+1} \Omega_b(v_i, v_k))
\]

\[ (2.3-14) \]

Here

\[
\Omega_b(v_k, v_i) = N_k g(v_k) \beta(v_k, v_i)
\]

\[ (2.3-15) \]

A more mathematically rigorous formulation is given by Ramakrishna [13], where the breakage rate is expressed as

\[
B_b(i) = \sum_{i}^{N} n_{i,k} g(v_k) N_k
\]

\[ (2.3-16) \]

where

\[
n_{i,k} = \int_{v_i}^{v_{i+1}} \frac{v_{i+1} - v}{v_{i+1} - v_i} \beta(v_k, v) dv + \int_{v_{i-1}}^{v_i} \frac{v - v_{i-1}}{v_i - v_{i-1}} \beta(v_k, v) dv
\]

\[ (2.3-17) \]

The Ramakrishna formulation can be slow due to the large number of integration points required. However, for simple forms of \( \beta \), the integrations can be performed relatively easily. The Hagesather formulation requires fewer integration points and the difference in accuracy with the Ramakrishna formulation can be corrected by a suitable choice of bin sizes.
To keep the computing time reasonable, a volume averaged value is used for the turbulent eddy dissipation when the Luo model is used in conjunction with the Ramakrishna formulation.

### 2.3.2 The Standard Method of Moments (SMM)

The SMM, proposed by Randolph and Larson \cite{26} is an alternative method for solving the PBE. Its advantages are that it reduces the dimensionality of the problem and that it is relatively simple to solve transport equations for lower-order moments. The disadvantages are that exact closure of the right-hand side is possible only in cases of constant aggregation and size-independent growth, and that breakage modeling is not possible. The closure constraint is overcome, however, through QMOM (see Section 2.3.3: The Quadrature Method of Moments (QMOM)).

#### Numerical Method

The SMM approach is based on taking moments of the PBE with respect to the internal coordinate (in this case, the particle size $L$).

Defining the $k$th moment as

$$ m_k(\vec{x}, t) = \int_0^\infty n(L; \vec{x}, t) L^k dL \quad k = 0, 1, \ldots, N - 1 $$

and assuming constant particle growth, its transport equation can be written as

$$ \frac{\partial}{\partial t}(\rho m_k) + \nabla \cdot (\rho \vec{u} m_k) = \rho(B_{ag,k} - D_{ag,k} + B_{br,k} - D_{br,k}) + \dot{n}_0 + \text{Growth} \quad (2.3-19) $$

where

$$ B_{ag,k} = \frac{1}{2} \int_0^\infty n(\lambda) \int_0^\infty a(u, \lambda)(u, \lambda)(u^3 + \lambda^3)^{k/3} n(u) du d\lambda $$

$$ D_{ag,k} = \int_0^\infty L^k n(L) \int_0^\infty a(L, \lambda)n(\lambda) d\lambda dL $$

$$ B_{br,k} = \int_0^\infty L^k \int_0^\infty g(\lambda) \beta(L | \lambda) n(\lambda) d\lambda dL $$

$$ D_{br,k} = \int_0^k L^k n(L) dL $$

$N$ is the specified number of moments and $\dot{n}_0$ is the nucleation rate. The growth term is defined as

$$ \text{Growth} \equiv \int_0^\infty kL^{k-1} G(L)n(L, t) dL $$

(2.3-24)
and for constant growth is represented as

\[ kGm_{k-1} \]  

Equation 2.3-20 can be derived by using

\[ u^3 = L^3 - \lambda^3; \quad dL = \frac{u^2}{L^2}du \]

and reversing the order of integration. From these moments, the parameters describing the gross properties of particle population can be derived as

\[
\begin{align*}
N_{\text{total}} &= m_0 \quad \text{(2.3-26)} \\
L_{\text{total}} &= m_1 \quad \text{(2.3-27)} \\
A_{\text{total}} &= K_a m_2 \quad \text{(2.3-28)} \\
V_{\text{total}} &= K_v m_3 \quad \text{(2.3-29)} \\
d_{32} &= \frac{m_3}{m_2} \quad \text{(2.3-30)}
\end{align*}
\]

These properties are related to the total number, length, area, and volume of solid particles per unit volume of mixture suspension. The Sauter mean diameter, \(d_{32}\), is usually used as the mean particle size.

To close Equation 2.3-19, the quantities represented in Equations 2.3-20–2.3-23 need to be expressed in terms of the moments being solved. To do this, one approach is to assume size-independent kernels for breakage and aggregation, in addition to other simplifications such as the Taylor series expansion of the term \((u^3 + \lambda^3)^{k/3}\). Alternatively, a profile of the PSD could be assumed so that Equations 2.3-20–2.3-23 can be integrated and expressed in terms of the moments being solved.

In ANSYS FLUENT, an exact closure is implemented by restricting the application of the SMM to cases with size-independent growth and a constant aggregation kernel.
2.3.3 The Quadrature Method of Moments (QMOM)

The quadrature method of moments (QMOM) was first proposed by McGraw [21] for modeling aerosol evolution and coagulation problems. Its applications by Marchisio et al. [19] have shown that the method requires a relatively small number of scalar equations to track the moments of population with small errors.

The QMOM provides an attractive alternative to the discrete method when aggregation quantities, rather than an exact PSD, are desired. Its advantages are fewer variables (typically only six or eight moments) and a dynamic calculation of the size bins. The disadvantages are that the number of abscissas may not be adequate to describe the PSD and that solving the Product-Difference algorithm may be time consuming.

Numerical Method

The quadrature approximation is based on determining a sequence of polynomials orthogonal to $n(L)$ (i.e., the particle size distribution). If the abscissas of the quadrature approximation are the nodes of the polynomial of order $N$, then the quadrature approximation

$$\int_0^\infty f(L)n(L)dL \approx \sum_{i=1}^{N} f(L_i)w_i, \quad (2.3-31)$$

is exact if $f(L)$ is a polynomial of order $N$ or smaller [5]. In all other cases, the closer $f(L)$ is to a polynomial, the more accurate the approximation.

A direct way to calculate the quadrature approximation is by means of its definition through the moments:

$$m_k = \sum_{i=1}^{N} w_i L_i^k. \quad (2.3-32)$$

The quadrature approximation of order $N$ is defined by its $N$ weights $w_i$ and $N$ abscissas $L_i$ and can be calculated by its first $2N$ moments $m_0, \ldots, m_{2N-1}$ by writing the recursive relationship for the polynomials in terms of the moments $m_k$. Once this relationship is written in matrix form, it is easy to show that the roots of the polynomials correspond to the eigenvalues of the Jacobi matrix [24]. This procedure is known as the Product-Difference algorithm [8]. Once the weights and abscissas are known, the source terms due to coalescence and breakage can be calculated and therefore the transport equations for the moments can be solved.
Applying Equations 2.3-31 and 2.3-32, the birth and death terms in Equation 2.3-19 can be rewritten as

\[
B_{ag,k} = \frac{1}{2} \sum_{i=1}^{N} w_i \sum_{j=1}^{N} w_j (L_i^3 + L_j^3)^{3/3} \alpha(L_i, L_j) \quad (2.3-33)
\]

\[
D_{ag,k} = \sum_{i=1}^{N} L_i^k w_i \sum_{j=1}^{N} w_j \alpha(L_i, L_j) \quad (2.3-34)
\]

\[
B_{br,k} = \sum_{i=1}^{N} w_i \int_{0}^{\infty} L_k g(L_i) \beta(L | L_i) dL \quad (2.3-35)
\]

\[
D_{br,k} = \sum_{i=1}^{N} w_i L_i^k g(L_i) \quad (2.3-36)
\]

Theoretically, there is no limitation on the expression of breakage and aggregation kernels when using QMOM.

The nucleation rate is defined in the same way as for the SMM. The growth rate for QMOM is defined by Equation 2.3-24 and represented as

\[
\sum_{i=1}^{N} w_i L_i^{k-1} G(L_i) \quad (2.3-37)
\]

to allow for a size-dependent growth rate.

### 2.4 Reconstructing the Particle Size Distribution from Moments

Given a set of moments, the most likely PSD can be obtained based on the “statistically most probable” distribution for turbulent flames [23], which was adapted for crystallization problems by Baldyga and Orciuch [3].

The number density function \(n(L)\) is expressed as

\[
n(L) = \exp \left( \sum_{i=0}^{N-1} A_i L^i \right) \quad (2.4-1)
\]

The equation for the \(k\)th moment is now written as

\[
m_k = \int_{0}^{\infty} L^k \exp \left( \sum_{i=0}^{N-1} A_i L^i \right) dL \quad k = 0, 1, \ldots, N - 1 \quad (2.4-2)
\]

Given \(N\) moments, the coefficients \(A_i\) can be found by a globally convergent Newton-Raphson method to reconstruct the particle size distribution (e.g., Figure 2.4.1).
Figure 2.4.1: Reconstruction of a Particle Size Distribution
Chapter 3. Using the ANSYS FLUENT Population Balance Model

This chapter provides basic instructions to install the population balance model and solve population balance problems in ANSYS FLUENT. It assumes that you are already familiar with standard ANSYS FLUENT features, including the user-defined function procedures described in the ANSYS FLUENT UDF Manual. This chapter describes the following:

- Section 3.1: Population Balance Module Installation
- Section 3.2: Loading the Population Balance Module
- Section 3.3: Population Balance Model Setup

3.1 Population Balance Module Installation

The population balance module is provided as an addon module with the standard ANSYS FLUENT licensed software.

3.2 Loading the Population Balance Module

The population balance module is loaded into ANSYS FLUENT through the text user interface (TUI). The module can only be loaded when a valid ANSYS FLUENT case file has been set or read. The text command to load the module is

\texttt{define \rightarrow models \rightarrow addon-module.}

A list of ANSYS FLUENT addon modules is displayed:

\texttt{FLUENT Addon Modules:}
0. none
1. MHD Model
2. Fiber Model
3. Fuel Cell and Electrolysis Model
4. SOFC Model with Unresolved Electrolyte
5. Population Balance Model
Enter Module Number: [0] 5
Select the Population Balance Model by entering the module number 5. During the loading process a scheme library containing the graphical and text user interface, and a UDF library containing a set of user defined functions are loaded into ANSYS FLUENT. A message Addon Module: popbal...loaded! is displayed at the end of the loading process.

The population balance module setup is saved with the ANSYS FLUENT case file. The module is loaded automatically when the case file is subsequently read into ANSYS FLUENT. Note that in the saved case file, the population balance module is saved with the absolute path. Therefore, if the locations of the population balance module installation or the saved case file are changed, ANSYS FLUENT will not be able to load the module when the case file is subsequently read.

### 3.3 Population Balance Model Setup

Following the loading of the population balance module, enable either the mixture or Eulerian multiphase model. This will allow you to activate the population balance model, where you will specify the appropriate parameters, and supply multiphase boundary conditions. These inputs are described in this chapter. Using the double-precision version of ANSYS FLUENT when solving population balance problems is highly recommended.

A limitation of the population balance model is that it can be used only on one secondary phase, even if your problem includes additional secondary phases. Note that a three-phase gas-liquid-solid case can be modeled, where the population balance model is used for the gas phase and the solid phase acts as a catalyst.

### 3.3.1 Enabling the Population Balance Model

The procedure for setting up a population balance problem is described below. (Note that this procedure includes only those steps necessary for the population balance model itself; you will need to set up other models, boundary conditions, etc. as usual. See the ANSYS FLUENT User’s Guide for details.)

1. Start the double-precision version of ANSYS FLUENT.
2. To enable the population balance model, follow the instructions in Section 3.2: Loading the Population Balance Module.
   
   Remember to enable the mixture or Eulerian multiphase model.
3. Open the Population Balance Model dialog box (Figure 3.3.1).

   ![Models → Population Balance → Edit...](image)

4. Specify the population balance method under Method.
3.3 Population Balance Model Setup

Figure 3.3.1: The Population Balance Model Dialog Box

- If you select **Discrete**, you will need to specify the following parameters:

  **Definition** can be specified as a **Geometric Ratio** or as a **File**. If **Geometric Ratio** is selected, then the **Ratio Exponent** must be specified. If **File** is selected, you will click the **Load File...** button and select the bin size file that you want loaded.

  You can input the diameter through the text file, with each diameter listed on a separate line, starting from the largest to the smallest diameter. Hence, you are not limited by the choices specified in the dialog box.

  **Number** specifies the number of particle size bins used in the calculation.

  **Ratio Exponent** specifies the exponent $q$ used in the discretization of the growth term volume coordinate (see Section 2.3.1: Numerical Method).

  **Min** specifies the minimum bin size $L_0 \equiv \left( \frac{V_0}{K_v} \right)^{1/3}$. 
Max displays the maximum bin size, which is calculated internally.

Kv specifies the value for the particle volume coefficient $K_v$ (as described in Section 2.2.1: Particle Growth and Dissolution). By default, this coefficient has a value of $\pi/6$.

To display a list of the bin sizes in the console window, click Print Bins. The bin sizes will be listed in order of size, from the largest to the smallest. This option is only available when the Geometric Ratio Definition is selected.

- If you select Standard Moment under Method, you will specify the number of Moments under Parameters.
- If you selected Quadrature Moment under Method, you will set the number of moments to either 4, 6 or 8 under Parameters.

5. Select the secondary phase from the Phase drop-down list for which you want to apply the population balance model parameters.

6. For all population balance methods, you can enable the following under Phenomena:

   **Nucleation Rate** allows you to specify the nucleation rate (particles/m$^3\cdot$s). You can select constant or user-defined from the drop-down list. If you select constant, specify a value in the adjacent field. If you have a user-defined function (UDF) that you want to use to model the nucleation rate, you can choose the user-defined option and specify the appropriate UDF.

   **Growth Rate** allows you to specify the particle growth rate (m/s). You can select constant or user-defined from the drop-down list. If you select constant, specify a value in the adjacent field. If you have a user-defined function (UDF) that you want to use to model the growth rate, you can choose the user-defined option and specify the appropriate UDF.

   **Aggregation Kernel** allows you to specify the aggregation kernel (m$^3$/s). You can select constant, luo-model, free-molecular-model, turbulent-model, or user-defined from the drop-down list:

   - If you select constant, specify a value in the adjacent field.
   - If you select luo-model, the Surface Tension for Population Balance dialog box will open automatically to allow you to specify the surface tension (see Figure 3.3.2). The aggregation rate for the model will then be calculated based on Luo’s aggregation kernel (as described in Section 2.2.2: Particle Birth and Death Due to Breakage and Aggregation).
   - If you select free-molecular-model, then Equation 2.2-33 is applied.
   - If you select turbulent-model, the Hamaker Constant for Population Balance dialog box will open automatically to allow you to specify the Hamaker constant (see Figure 3.3.3). More information about this model is available in Section 2.2.2: Turbulent Aggregation Kernel.
3.3 Population Balance Model Setup

**Figure 3.3.2: The Surface Tension for Population Balance Dialog Box**

**Figure 3.3.3: The Hamaker Constant for Population Balance Dialog Box**

- If you have a user-defined function (UDF) that you want to use to model the aggregation rate, you can choose the **user-defined** option and specify the appropriate UDF.

**Breakage Kernel** allows you to specify the particle breakage frequency (particles/m$^3$.s). You can select **constant**, luo-model, lehr-model, ghadiri-model, or user-defined from the **Frequency** drop-down list:

- If you select **constant**, specify a value in the adjacent field.
- If you select **luo-model**, the Surface Tension for Population Balance dialog box will open automatically to allow you to specify the surface tension (see Figure 3.3.2). The frequency used in the breakage rate will then be calculated based on Luo’s breakage kernel (as described in Section 2.2.2: Particle Birth and Death Due to Breakage and Aggregation).
- If you select **lehr-model**, the Surface Tension and Weber Number dialog box will open automatically to allow you to specify the surface tension and critical Weber number (see Figure 3.3.4). The frequency used in the breakage rate will then be calculated based on Lehr’s breakage kernel (as described in Section 2.2.2: Particle Birth and Death Due to Breakage and Aggregation).
- If you select **ghadiri-model**, the Ghadiri Breakage Constant for Population Balance dialog box will open automatically to allow you to specify the breakage constant (see Figure 3.3.5). The frequency will then be calculated based on Ghadiri’s breakage kernel (as described in Section 2.2.2: Particle Birth and Death Due to Breakage and Aggregation).
- If you have a user-defined function (UDF) that you want to use to model
Using the ANSYS FLUENT Population Balance Model

Figure 3.3.4: The Surface Tension and Weber Number Dialog Box

Figure 3.3.5: The Ghadiri Breakage Constant for Population Balance Dialog Box

the frequency for the breakage rate, you can choose the user-defined option and specify the appropriate UDF.
If you selected constant, ghadiri-model, or user-defined for Frequency, then you can specify the probability density function used to calculate the breakage rate by making a selection in the PDF drop-down list. You can select parabolic, generalized, or user-defined:

- If you select parabolic, the Shape Factor for Parabolic PDF dialog box will open automatically to allow you to specify the shape factor $C$ (see Figure 3.3.6). The PDF used in the breakage rate will then be calculated according to Equation 2.2-15 (as described in Section 2.2.2: Particle Birth and Death Due to Breakage and Aggregation).

![Shape Factor for Parabolic PDF Dialog Box](image)

Figure 3.3.6: The Shape Factor for Parabolic PDF Dialog Box

- If you select generalized, the Generalized pdf for multiple breakage dialog box will open automatically (Figure 3.3.7).

![Generalized pdf for multiple breakage Dialog Box](image)

Figure 3.3.7: The Generalized pdf for multiple breakage Dialog Box
Perform the following steps in the Generalized pdf for multiple breakage dialog box:

(a) Select either One Term or Two Term from the Options list. Your selection will determine whether \(i\) in Equation 2.2-21 is 0 or 1, respectively.

(b) Enter a value for the averaged Number of Daughters. It can be any real number (including non-integers, e.g., 2.5), as long as it is not less than 2.

(c) Define the parameter(s) for Equation 2.2-21 in the Input Parameters group box. When One Term is selected from the Options list, you must enter a value for \(q_{i0}\). When Two Term is selected from the Options list, you must enter values for \(w_{i0}\), \(p_{i0}\), \(q_{i0}\), \(r_{i0}\), and \(q_{i1}\). For information about appropriate values for these parameters to result in the daughter distributions shown in Table 2.2.2, see Table 2.2.3.

(d) Click the Validate/Apply button to save the settings. The text boxes in the All Parameters group box will be updated, using the values you entered in the Input Parameters group box, as well as values derived from the constraints shown in Equations 2.2-22–2.2-24.

(e) Verify that the values in the All Parameters group box represent your intended PDF before clicking Close.

- If you have a user-defined function (UDF) that you want to use to model the PDF for the breakage rate, you can choose the user-defined option and specify the appropriate UDF. See Chapter 5: UDFs for Population Balance Modeling for details about UDFs for the population balance model.

Choose between the default Hagesather formulation and the Ramakrishna formulation. Detailed information about these two methods can be found in Section 2.3.1: Breakage Formulations for the Discrete Method.

7. Specify the boundary conditions for the solution variables.
   - [Boundary Conditions]

   See Section 3.3.2: Defining Population Balance Boundary Conditions below.

8. Specify the initial guess for the solution variables.
   - [Solution Initialization]

9. Solve the problem and perform relevant postprocessing functions.
   - [Run Calculation]

   See Chapter 4: Postprocessing for the Population Balance Model for details about postprocessing.
3.3.2 Defining Population Balance Boundary Conditions

To define boundary conditions specific to the population balance model, use the following procedure:

1. In the **Boundary Conditions** task page, select the secondary phase(s) in the **Phase** drop-down list and then open the appropriate boundary condition dialog box (e.g., Figure 3.3.8).

![Boundary Conditions](image)

Figure 3.3.8: Specifying Inlet Boundary Conditions for the Population Balance Model

2. In the **Multiphase** tab, under **Boundary Condition**, select the type of boundary condition for each bin (for the discrete method) or moment (for SMM and QMOM) as either **Specified Value** or **Specified Flux**.

Note that the boundary condition variables (e.g., Bin-0) are labeled according to the following:

\[ \text{bin/moment} - \text{ith bin/moment} \]

where the \( i \)th *bin/moment* can range from 0 (the first bin or moment) to \( N - 1 \), where \( N \) is the number of bins/moments that you entered in the **Population Balance Model** dialog box.
3. Under Population Balance Boundary Value, enter a value or a flux as appropriate.

- If you selected Specified Value for the selected boundary variable, enter a value in the field adjacent to the variable name. This value will correspond to the variable $f_i$ in Equation 2.3-4 (for the discrete method) or $m_k$ in Equation 2.3-19 (for SMM or QMOM).
- If you selected Specified Flux for the selected boundary variable, enter a value in the field adjacent to the variable name. This value will be the spatial particle volume flux $dV/dx_i$.

### 3.3.3 Specifying Population Balance Solution Controls

In the Equations dialog box (Figure 3.3.9), equations for each bin (e.g., phase-2 Bin) will appear in the Equations list.

![Solution Controls](Equations...)

The default value under Under-Relaxation Factors (in the Solution Controls task page) for the population balance equations is 0.5, and the default Discretization scheme (in the Solution Methods task page) is First Order Upwind.

![Figure 3.3.9: The Equations Dialog Box](Equations_Dialog_Box)
3.3.4 Coupling With Fluid Dynamics

To couple population balance modeling of the secondary phase(s) with the overall problem fluid dynamics, a Sauter mean diameter ($d_{32}$ in Equation 2.3-30) may be used to represent the particle diameter of the secondary phase. The Sauter mean diameter is defined as the ratio of the third moment to the second moment for the SMM and QMOM. For the discrete method, it is defined as

$$d_{32} = \frac{\sum N_i L_i^3}{\sum N_i L_i^2}$$

(3.3-1)

To specify the Sauter mean diameter as the secondary phase particle diameter, open the Secondary Phase dialog box.

![Secondary Phase Dialog Box](image)

In the Secondary Phase dialog box (e.g., Figure 3.3.10), select sauter-mean from the Diameter drop-down list under Properties. Note that a constant diameter or user-defined function may also be used.

Figure 3.3.10: The Secondary Phase Dialog box for Hydrodynamic Coupling
### 3.3.5 Specifying Interphase Mass Transfer Due to Nucleation and Growth

In applications that involve the creation, dissolution, or growth of particles (e.g., crystallization), the total volume fraction equation for the particulate phase will have source terms due to these phenomena. The momentum equation for the particulate phase will also have source terms due to the added mass. In ANSYS FLUENT, the mass source term can be specified using the UDF hook `DEFINE_HET_RXN_RATE`, as described in Appendix A: `DEFINE_HET_RXN_RATE` Macro, or using the Phase Interaction dialog box, described below.

As an example, in crystallization, particles are created by means of nucleation (\(\dot{n}_0\)), and a growth rate (\(G\)) can also be specified. The mass transfer rate of formation (in kg/m\(^3\)-s) of particles of all sizes is then

\[
\dot{m} = 3\rho K_v \int_0^\infty L^2 G n(L) dL = \frac{1}{2} \rho K_a \int_0^\infty L^2 G n(L) dL
\]

For the discrete method, the mass transfer rate due to growth can be written as

\[
\dot{m} = \rho \int_0^\infty G_v n(L) dL = \rho \int_0^\infty G_v n(V) dV = \rho \sum_i G_{v,i} N_i
\]

If the nucleation rate is included in the total mass transfer, then the mass transfer becomes

\[
\dot{m} = \rho V_0 \dot{n}_0 + \sum_i \rho G_{v,i} N_i
\]

For the discrete method, the sources to the population balance equations must sum to the total mass transfer rate. To access the sources, you can use the macro `C_PB_DISCI_PS (cell, thread, i)`.

See Chapter 5: UDFs for Population Balance Modeling for more information about macros for population balance variables.
For the SMM, only a size-independent growth rate is available. Hence, the mass transfer rate can be written as

$$\dot{m} = \frac{1}{2} \rho K_a G m_2$$  \hspace{1cm} (3.3-5)

For the QMOM, the mass transfer rate can be written as

$$\dot{m} = \frac{1}{2} \rho K_a \sum_i L_i^2 w_i G(L_i)$$  \hspace{1cm} (3.3-6)

For both the SMM and QMOM, mass transfer due to nucleation is negligible, and is not taken into account.

Note that for crystallization, the primary phase is comprised of multiple components. At the very least, there is a solute and a solvent. To define the multicomponent multiphase system, you will need to activate Species Transport in the Species Model dialog box for the primary phase after activating the multiphase model. The rest of the procedure for setting up a species transport problem is identical to setting up species in single phase. The heterogeneous reaction is defined as:

Solute (liquid) $\rightarrow$ Crystal (secondary phase)

When the population balance model is activated, mass transfer between phases for non-reacting species (such as boiling) and heterogeneous reactions (such as crystallization) can be done automatically, in lieu of hooking a UDF.

For simple unidirectional mass transfer between primary and secondary phases due to nucleation and growth phenomena of non-reacting species, go to the Phases task page and click the Interaction... button. This will open the Phase Interaction dialog box (Figure 3.3.11). Click the Mass tab to specify the Mass Transfer of species between the phases. Specify the Number of Mass Transfer Mechanisms involved in your case. From the drop-down list under From Phase, select the phase that you want to transfer mass from. In the To Phase drop-down list, select the phase that you want to transfer mass to.
You have a choice of four mechanisms used to transfer mass. Under Mechanism select from the drop-down list

- none if you do not want any mass transfer between the phases.
- constant-rate for a fixed, user-specified rate.
- user-defined if you hooked a UDF describing the mass transfer mechanism.
- population-balance for an automated method of mass transfer, not involving a UDF.

Click OK to save the settings.

For heterogeneous reactions, the Species Transport model has to be activated for the primary phase. In the Phases task page, click the Interaction... button. This will open the Phase Interaction dialog box (Figure 3.3.12). Click the Reactions tab to specify the stoichiometry for the reactant and the product. At the bottom of the Phase Interaction dialog box, select population-balance as the Reaction Rate Function. Click OK to save the settings. Either this method or the use of the UDF, described in Appendix A: DEFINE_HET_RXN_RATE Macro, will produce the same results.

![Phase Interaction Dialog Box](image_url)

Figure 3.3.11: The Phase Interaction Dialog Box for Non-reacting Species
Figure 3.3.12: The Phase Interaction Dialog Box for a Heterogeneous Reaction
Chapter 4. Postprocessing for the Population Balance Model

ANSYS FLUENT provides postprocessing options for displaying, plotting, and reporting various particle quantities, which include the main solution variables and other auxiliary quantities.

- Section 4.1: Population Balance Solution Variables
- Section 4.2: Reporting Derived Population Balance Variables

4.1 Population Balance Solution Variables

Solution variables that can be reported for the population balance model are:

- Bin-i fraction (discrete method only), where i is N-1 bins/moments.
- Number density of Bin-i fraction (discrete method only)
- Diffusion Coef. of Bin-i fraction/Moment-i
- Sources of Bin-i fraction/Moment-i
- Moment-i (SMM and QMOM only)
- Abscissa-i (QMOM method only)
- Weight-i (QMOM method only)

Bin-i fraction is the fraction \( f_i \) of the volume fraction for the \( i \)th size bin when using the discrete method. Number density of Bin-i fraction is the number density \( N_i \) in particles/m\(^3\) for the \( i \)th size bin. Moment-i is the \( i \)th moment of the distribution when using the standard method of moments or the quadrature method of moments.

Though the diffusion coefficients of the population variables are available (e.g., Diffusion Coef. of Bin-i fraction/Moment-i), they are set to zero because the diffusion term is not present in the population balance equations.
4.2 Reporting Derived Population Balance Variables

Two options are available in the Report menu that allow you to report computed moments and number density on selected surfaces or cell zones of the domain.

4.2.1 Computing Moments

You can compute moments for the population balance model using the Population Balance Moments dialog box (Figure 4.2.1).

![Figure 4.2.1: The Population Balance Moments Dialog Box](image)

The steps for computing moments are as follows:

1. For the discrete method, specify the Number of Moments. For the SMM and QMOM, the number of moments is set equal to the number of moments that were solved, and thus cannot be changed.

2. For a surface average, select the surface(s) on which to calculate the moments in the Surfaces list.

3. For a volume average, select the volume(s) in which to calculate the moments in the Cell Zones list.

4. Click Print to display the moment values in the console window.

5. To save the moment calculations to a file, click Write... and enter the appropriate information in the resulting Select File dialog box. The file extension should be .pb.
4.2 Reporting Derived Population Balance Variables

4.2.2 Displaying a Number Density Function

You can display the number density function for the population balance model using the Number Density Function dialog box (Figure 4.2.2).

![Number Density Function](image)

Figure 4.2.2: The Number Density Function Dialog Box

The steps for displaying the number density function are as follows:

1. Specify the **Report Type** as either a **Surface Average** or a **Volume Average**.

2. Under **Plot Type**, specify how you would like to display the number density function data.
   - **Histogram** displays a histogram of the discrete number density ($N_i$). The number of divisions in the histogram is equal to the number of bins specified in the Population Balance Model dialog box. This option is available only with the discrete method.
   - **Curve** displays a smooth curve of the number density function.

3. In the **Fields** list, select the data to be plotted.
   - **Discrete Number Density** ($N_i$) is the number of particles per unit volume of physical space in the $i$th size bin plotted against particle diameter size $i$. This option is available only with the discrete method.
   - **Length Number Density Function** ($n(L)$) is the number of particles per unit volume of physical space per unit particle length plotted against particle diameter.
   - **Volume Number Density Function** ($n(V)$) is the number of particles per unit volume of physical space per unit particle volume plotted against particle volume.
4. Choose the cell zones on which to plot the number density function data in the Cell Zones list.

5. Click Plot... to display the data.

6. (optional) Click Print to display the number density function data in the console window.

7. Click Write to save the number density function data to a file. The Select File dialog box will open, where you can specify a name and save a file containing the plot data. The file extension should be .pbd.
Chapter 5. UDFs for Population Balance Modeling

5.1 Population Balance Variables

The macros listed in Table 5.1.1 can be used to return real variables associated with the population balance model. The variables are available in both the pressure-based and density-based solvers. The macros are defined in the `sg_pb.h` header file, which is included in `udf.h`.

Table 5.1.1: Macros for Population Balance Variables Defined in `sg_pb.h`

<table>
<thead>
<tr>
<th>Macro</th>
<th>Argument Types</th>
<th>Returns</th>
</tr>
</thead>
<tbody>
<tr>
<td>C_PB_DISCI</td>
<td>cell_t c, Thread *t, int i</td>
<td>fraction (f_i) of the total volume fraction for the (i)th size bin</td>
</tr>
<tr>
<td>C_PB_SMMI</td>
<td>cell_t c, Thread *t, int i</td>
<td>(i)th moment</td>
</tr>
<tr>
<td>C_PB_QMOMI</td>
<td>cell_t c, Thread *t, int i</td>
<td>(i)th moment, where (i = 0, 1, 2, 3, 4, 5)</td>
</tr>
<tr>
<td>C_PB_QMOMI_L</td>
<td>cell_t c, Thread *t, int i</td>
<td>abscissa (L_i), where (i = 0, 1, 2)</td>
</tr>
<tr>
<td>C_PB_QMOMI_W</td>
<td>cell_t c, Thread *t, int i</td>
<td>weight (w_i), where (i = 0, 1, 2)</td>
</tr>
<tr>
<td>C_PB_DISCI_PS</td>
<td>cell_t c, Thread *t, int i</td>
<td>net source term to (i)th size bin</td>
</tr>
<tr>
<td>C_PB_SMMI_PS</td>
<td>cell_t c, Thread *t, int i</td>
<td>net source term to (i)th moment</td>
</tr>
<tr>
<td>C_PB_QMOMI_PS</td>
<td>cell_t c, Thread *t, int i</td>
<td>net source term to (i)th moment</td>
</tr>
</tbody>
</table>

5.2 Population Balance DEFINE Macros

This section contains descriptions of DEFINE macros for the population balance model. Definitions of each DEFINE macro are contained in the `udf.h` header file.

- DEFINE_PB_BREAKUP_RATE_FREQ
- DEFINE_PB_BREAKUP_RATE_PDF
- DEFINE_PB_COALESCENCE_RATE
- DEFINE_PB_NUCLEATION_RATE
- DEFINE_PB_GROWTH_RATE
5.2.1 DEFINE_PB_BREAKUP_RATE_FREQ

You can use the DEFINE_PB_BREAKUP_RATE_FREQ macro if you want to define the breakage frequency using a UDF. The function is executed at the beginning of every time step.

Usage

```
DEFINE_PB_BREAKUP_RATE_FREQ(name, cell, thread, d_1)
```

<table>
<thead>
<tr>
<th>Argument Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>char name</td>
<td>UDF name</td>
</tr>
<tr>
<td>cell_t cell</td>
<td>Cell index</td>
</tr>
<tr>
<td>Thread *thread</td>
<td>Pointer to the secondary phase thread</td>
</tr>
<tr>
<td>real d_1</td>
<td>Parent particle diameter or length</td>
</tr>
</tbody>
</table>

Function returns
real

There are four arguments to DEFINE_PB_BREAKUP_RATE_FREQ: name, cell, thread, and d_1. You will supply name, the name of the UDF. cell, thread, and d_1 are variables that are passed by the ANSYS FLUENT solver to your UDF.

Example

Included below is an example of a UDF that defines a breakage frequency (see Section 2.2.2: Particle Birth and Death Due to Breakage and Aggregation) that is based on the work of Tavlarides [4], such that

\[
g(V') = C_1 \frac{\varepsilon^{1/3}}{(1 + \alpha)d^{2/3}} \exp \left( -C_2 \frac{\sigma(1 + \alpha)^2}{\rho_l \varepsilon^{2/3} d^{5/3}} \right)
\]  

(5.2-1)

where \(C_1\) and \(C_2\) are constants, \(\varepsilon\) is the dissipation rate, \(d\) is the parent diameter, \(\sigma\) is the surface tension, \(\alpha\) is the volume fraction of the dispersed phase, and \(\rho_l\) is the density of the primary phase.
5.2 Population Balance DEFINE Macros

UDF that computes the particle breakage frequency

```c
#include "udf.h"
#include "sg_pb.h"
#include "sg_mphase.h"

DEFINE_PB_BREAKUP_RATE_FREQ(break_up_freq_tav, cell, thread, d_1)
{
    real epsi, alpha, f1, f2, rho_d;
    real C1 = 0.00481, C2 = 0.08, sigma = 0.07;
    Thread *tm = THREAD_SUPER_THREAD(thread); /* passed thread is phase */
    epsi = C_D(cell, tm);
    alpha = C_VOF(cell, thread);
    rho_d = C_R(cell, thread);
    f1 = pow(epsi, 1./3.)/(1.+epsi)*pow(d_1, 2./3.));
    f2 = -(C2*sigma*(1.+epsi))/(rho_d*pow(epsi,2./3.)*pow(d_1, 5./3.));
    return C1*f1*exp(f2);
}
```

5.2.2 DEFINE_PB_BREAKUP_RATE_PDF

You can use the DEFINE_PB_BREAKUP_RATE_PDF macro if you want to define the breakage PDF using a UDF. The function is executed at the beginning of every time step.

Usage

```c
DEFINE_PB_BREAKUP_RATE_PDF(name, cell, thread, d_1, d_2)
```

<table>
<thead>
<tr>
<th>Argument Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>char name</td>
<td>UDF name</td>
</tr>
<tr>
<td>cell_t cell</td>
<td>Cell index</td>
</tr>
<tr>
<td>Thread *thread</td>
<td>Pointer to the secondary phase thread</td>
</tr>
<tr>
<td>real d_1</td>
<td>Parent particle diameter or length</td>
</tr>
<tr>
<td>real d_2</td>
<td>Diameter of one of the daughter particles after breakage; the second daughter particle diameter is calculated by conservation of particle volume</td>
</tr>
</tbody>
</table>

Function returns

real
There are five arguments to DEFINE_PB_BREAKUP_RATE_FREQ: name, cell, thread, d_1, and d_2. You will supply name, the name of the UDF. cell, thread, d_1, and d_2 are variables that are passed by the ANSYS FLUENT solver to your UDF.

**Example**

Included below is an example of a UDF that defines a breakage PDF (see Section 2.2.2: Particle Birth and Death Due to Breakage and Aggregation) that is parabolic, as defined in Equation 2.2-15.

```plaintext
/************************************************************************
UDF that computes the particle breakage PDF
************************************************************************/

#include "udf.h"
#include "sg_pb.h"
#include "sg_mphase.h"

DEFINE_PB_BREAKUP_RATE_PDF(break_up_pdf_par, cell, thread, d_1, d_2)
{
  real pdf;
  real kv = M_PI/6.;

  real C = 1.0;

  real f_2, f_3, f_4;
  real V_prime = kv*pow(d_1,3.);
  real V = kv*pow(d_2,3.);

  f_2 = 24.*pow(V/V_prime,2.);
  f_3 = -24.*(V/V_prime);
  f_4 = 6.;

  pdf = (C/V_prime) + ((1.-C/2.)/V_prime)*(f_2 + f_3 + f_4);

  return 0.5*pdf;
}

5.2.3 DEFINE_PB_COALESCENCE_RATE

You can use the DEFINE_PB_COALESCENCE_RATE macro if you want to define your own particle aggregation kernel. The function is executed at the beginning of every time step.
5.2 Population Balance DEFINE Macros

Usage

DEFINE_PBCOALESCENCE_RATE(name, cell, thread, d_1, d_2)

<table>
<thead>
<tr>
<th>Argument Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>char name</td>
<td>UDF name</td>
</tr>
<tr>
<td>cell_t cell</td>
<td>Cell index</td>
</tr>
<tr>
<td>Thread *thread</td>
<td>Pointer to the secondary phase thread</td>
</tr>
<tr>
<td>real d_1, d_2</td>
<td>Diameters of the two colliding particles</td>
</tr>
</tbody>
</table>

Function returns
real

There are five arguments to DEFINE_PBCOALESCENCE_RATE: name, cell, thread, d_1, and d_2. You will supply name, the name of the UDF. cell, thread, d_1, and d_2 are variables that are passed by the ANSYS FLUENT solver to your UDF. Your UDF will need to return the real value of the aggregation rate.

Example

Included below is an example UDF for a Brownian aggregation kernel. In this example, the aggregation rate is defined as

$$a(L, \lambda) = a(V, V') = \beta_0 \frac{(L + \lambda)^2}{L\lambda}$$

where $\beta_0 = 1 \times 10^{-17}$ m$^3$/s.
UDFs for Population Balance Modeling

UDF that computes the particle aggregation rate
*************************************************************************
#include "udf.h"
#include "sg_pb.h"
#include "sg_mphase.h"

DEFINE_PB_COALESCENCE_RATE(aggregation_kernel,cell,thread,d_1,d_2)
{
    real agg_kernel;
    real beta_0 = 1.0e-17 /* aggregation rate constant */
    agg_kernel = beta_0*pow((d_1+d_2),2.0)/(d_1*d_2);
    return agg_kernel;
}

5.2.4 DEFINE_PB_NUCLEATION_RATE

You can use the DEFINE_PB_NUCLEATION_RATE macro if you want to define your own particle nucleation rate. The function is executed at the beginning of every time step.

Usage

DEFINE_PB_NUCLEATION_RATE(name, cell, thread)

Argument Type  Description
char name       UDF name
cell_t cell     Cell index
Thread *thread  Pointer to the secondary phase thread

Function returns
real

There are three arguments to DEFINE_PB_NUCLEATION_RATE: name, cell, and thread. You will supply name, the name of the UDF. cell and thread are variables that are passed by the ANSYS FLUENT solver to your UDF. Your UDF will need to return the real value of the nucleation rate.
Example

Potassium chloride can be crystallized from water by cooling. Its solubility decreases linearly with temperature. Assuming power-law kinetics for the nucleation rate,

\[ \dot{n}_0 = K_n (S - 1)^{N_n} \]

where \( K_n = 4 \times 10^{10} \) particles/m^3-s and \( N_n = 2.77 \).

UDF that computes the particle nucleation rate

```c
#include "udf.h"
#include "sg_pb.h"
#include "sg_mphase.h"

DEFINE_PB_NUCLEATION_RATE(nuc_rate, cell, thread)
{
    real J, S;
    real Kn = 4.0e10; /* nucleation rate constant */
    real Nn = 2.77; /* nucleation law power index */
    real T,solute_mass_frac,solvent_mass_frac, solute_mol_frac,solubility;
    real solute_mol_wt, solvent_mol_wt;

    Thread *tc = THREADSUPER_THREAD(thread); /*obtain mixture thread */
    Thread **pt = THREADSUB_THREADS(tc); /* pointer to sub_threads */
    Thread *tp = pt[P_PHASE]; /* primary phase thread */

    solute_mol_wt = 74.55; /* molecular weight of potassium chloride */
    solvent_mol_wt = 18.; /* molecular weight of water */
    solvent_mass_frac = C_YI(cell,tp,0);
    /* mass fraction of solute in primary phase (solvent) */

    solvent_mass_frac = 1.0 - solute_mass_frac;
    solute_mol_frac = (solute_mass_frac/solute_mol_wt)/
    ((solute_mass_frac/solute_mol_wt)+(solvent_mass_frac/solvent_mol_wt));

    T = C_T(cell,tp); /* Temperature of primary phase in Kelvin */

    solubility = 0.0005*T-0.0794;
    /* Solubility Law relating equilibrium solute mole fraction to Temperature*/
```
S = solute_mol_frac/solubility; /* Definition of Supersaturation */

if (S <= 1.)
{
    J = 0.;
}
else
{
    J = Kn*pow((S-1),Nn);
}
return J;

Note that the solubility and the chemistry could be defined in a separate
routine and simply called from the above function.

5.2.5 DEFINE_PB_GROWTH_RATE

You can use the DEFINE_PB_GROWTH_RATE macro if you want to define your own particle
growth rate. The function is executed at the beginning of every time step.

Usage

DEFINE_PB_GROWTH_RATE(name, cell, thread,d_i)

<table>
<thead>
<tr>
<th>Argument Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>char name</td>
<td>UDF name</td>
</tr>
<tr>
<td>cell_t cell</td>
<td>Cell index</td>
</tr>
<tr>
<td>Thread *thread</td>
<td>Pointer to the secondary phase thread</td>
</tr>
<tr>
<td>real d_i</td>
<td>Particle diameter or length</td>
</tr>
</tbody>
</table>

Function returns
real

There are four arguments to DEFINE_PB_GROWTH_RATE: name, cell, thread, and d_i. You will supply name, the name of the UDF. cell, thread, and d_i are variables that are passed by the ANSYS FLUENT solver to your UDF. Your UDF will need to return the real value of the growth rate.
5.2 Population Balance 

Example

Potassium chloride can be crystallized from water by cooling. Its solubility decreases linearly with temperature. Assuming power-law kinetics for the growth rate,

\[ G = K_g (S - 1)^{N_g} \]

where \( K_g = 2.8 \times 10^{-8} \text{ m/s} \) and \( N_g = 1 \).

UDF that computes the particle growth rate

```c
#include "udf.h"
#include "sg_pb.h"
#include "sg_mphase.h"

DEFINE_PB_GROWTH_RATE(growth_rate, cell, thread,d_1)
{
    /* d_1 can be used if size-dependent growth is needed */
    /* When using SMM, only size-independent or linear growth is allowed */

    real G, S;
    real Kg = 2.8e-8; /* growth constant */
    real Ng = 1.; /* growth law power index */
    real T,solute_mass_frac,solvent_mass_frac, solute_mol_frac,solubility;
    real solute_mol_wt, solvent_mol_wt;

    Thread *tc = THREAD_SUPER_THREAD(thread); /*obtain mixture thread */
    Thread **pt = THREAD_SUB_THREADS(tc); /* pointer to sub_threads */
    Thread *tp = pt[P_PHASE]; /* primary phase thread */

    solute_mol_wt = 74.55; /* molecular weight of potassium chloride */
    solvent_mol_wt = 18.; /* molecular weight of water */
    solute_mass_frac = C_YI(cell,tp,0);
    /* mass fraction of solute in primary phase (solvent) */

    solvent_mass_frac = 1.0 - solute_mass_frac;
    solute_mol_frac = (solute_mass_frac*solute_mol_wt)/(solute_mass_frac*solute_mol_wt+solvent_mass_frac*solvent_mol_wt);

    T = C_T(cell,tp); /* Temperature of primary phase in Kelvin */
```
solubility = 0.0005*T-0.0794; /* Solubility Law relating equilibrium solute mole fraction to Temperature*/

S = solute_mol_frac/solubility; /* Definition of Supersaturation */

if (S <= 1.)
{
    G = 0.;
}
else
{
    G = Kg*pow((S-1),Ng);
}
return G;

Note that the solubility and the chemistry could be defined in a separate routine and simply called from the above function.

5.3 Hooking a Population Balance UDF to ANSYS FLUENT

After the UDF that you have defined using DEFINE_PB_BREAKUP_RATE_FREQ, DEFINE_PB_BREAKUP_RATE_PDF, DEFINE_PB_COALESCENCE_RATE, DEFINE_PB_NUCLEATION_RATE, or DEFINE_PB_GROWTH_RATE is interpreted or compiled, the name that you specified in the DEFINE macro argument (e.g., agg_kernel) will become visible and selectable in the appropriate drop-down list under Phenomena in the Population Balance Model dialog box (Figure 3.3.1).
A.1 Description

You need to use DEFINE_HET_RXN_RATE to specify reaction rates for heterogeneous reactions. A heterogeneous reaction is one that involves reactants and products from more than one phase. Unlike DEFINE_VR_RATE, a DEFINE_HET_RXN_RATE UDF can be specified differently for different heterogeneous reactions.

During ANSYS FLUENT execution, the DEFINE_HET_RXN_RATE UDF for each heterogeneous reaction that is defined is called in every fluid cell. ANSYS FLUENT will use the reaction rate specified by the UDF to compute production/destruction of the species participating in the reaction, as well as heat and momentum transfer across phases due to the reaction.

A heterogeneous reaction is typically used to define reactions involving species of different phases. The bulk phase can participate in the reaction if the phase does not have any species (i.e. phase has fluid material instead of mixture material). Heterogeneous reactions are defined in the Phase Interaction dialog box.
A.2 Usage

DEFINE_HET_RXN_RATE(name, c, t, r, mw, yi, rr, rr_t)

<table>
<thead>
<tr>
<th>Argument Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>char name</td>
<td>UDF name.</td>
</tr>
<tr>
<td>cell_t c</td>
<td>Cell index.</td>
</tr>
<tr>
<td>Thread *t</td>
<td>Cell thread (mixture level) on which heterogeneous reaction rate is to be applied.</td>
</tr>
<tr>
<td>Hetero_Reaction *r</td>
<td>Pointer to data structure that represents the current heterogeneous reaction (see sg_mphase.h).</td>
</tr>
<tr>
<td>real mw[MAX_PHASES][MAX_SPE_EQNS]</td>
<td>Matrix of species molecular weights. mw[i][j] will give molecular weight of species with ID j in phase with index i. For phase which has fluid material, the molecular weight can be accessed as mw[i][0].</td>
</tr>
<tr>
<td>real yi[MAX_PHASES][MAX_SPE_EQNS]</td>
<td>Matrix of species mass fractions. yi[i][j] will give molecular weight of species with ID j in phase with index i. For phase which has fluid material, yi[i][0] will be 1.</td>
</tr>
<tr>
<td>real *rr</td>
<td>Pointer to laminar reaction rate.</td>
</tr>
<tr>
<td>real *rr_t</td>
<td>Currently not used. Provided for future use.</td>
</tr>
</tbody>
</table>

Function returns
void

There are eight arguments to DEFINE_HET_RXN_RATE: name, c, t, r, mw, yi, rr, and rr_t. You will supply name, the name of the UDF. c, t, r, mw, yi, rr, and rr_t are variables that are passed by the ANSYS FLUENT solver to your UDF. Your UDF will need to set the values referenced by the real pointer rr.
A.3 Example

The following compiled UDF, named arrh, defines an Arrhenius-type reaction rate. The rate exponents are assumed to be same as the stoichiometric coefficients.

```c
#include "udf.h"

static const real Arrhenius = 1.e15;
static const real E_Activation = 1.e6;
#define SMALL_S 1.e-29

DEFINE_HET_RXN_RATE(arrh,c,t,hr,mw,yi,rr,rr_t)
{
    Domain **domain_reactant = hr->domain_reactant;
    real *stoich_reactant = hr->stoich_reactant;
    int *reactant = hr->reactant;
    int i;
    int sp_id;
    int dindex;
    Thread *t_reactant;
    real ci;
    real T = 1200.; /* should obtain from cell */

    /* instead of compute rr directly, compute log(rr) and then 
    take exp */

    *rr = 0;
    for (i=0; i < hr->n_reactants; i++)
    {
        sp_id = reactant[i]; /* species ID to access mw and yi */

        if (sp_id == -1) sp_id = 0; /* if phase does not have species, 
                                  mw, etc. will be stored at index 0 */

        dindex = DOMAIN_INDEX(domain_reactant[i]);
                /* domain index to access mw & yi */

        t_reactant = THREAD_SUB_THREAD(t,dindex);

        /* get conc. */
        ci = yi[dindex][sp_id]*C_R(c,t_reactant)/mw[dindex][sp_id];

        ci = MAX(ci,SMALL_S);
```
A.4 Hooking a Heterogeneous Reaction Rate UDF to ANSYS FLUENT

After the UDF that you have defined using DEFINE_HET_RXN_RATE is interpreted or compiled (see the Fluent UDF Manual for details), the name that you specified in the DEFINE macro argument (e.g., arrh) will become visible and selectable under Reaction Rate Function in the Reactions tab of the Phase Interaction dialog box. (Note you will first need to specify the Total Number of Reactions greater than 0.)


