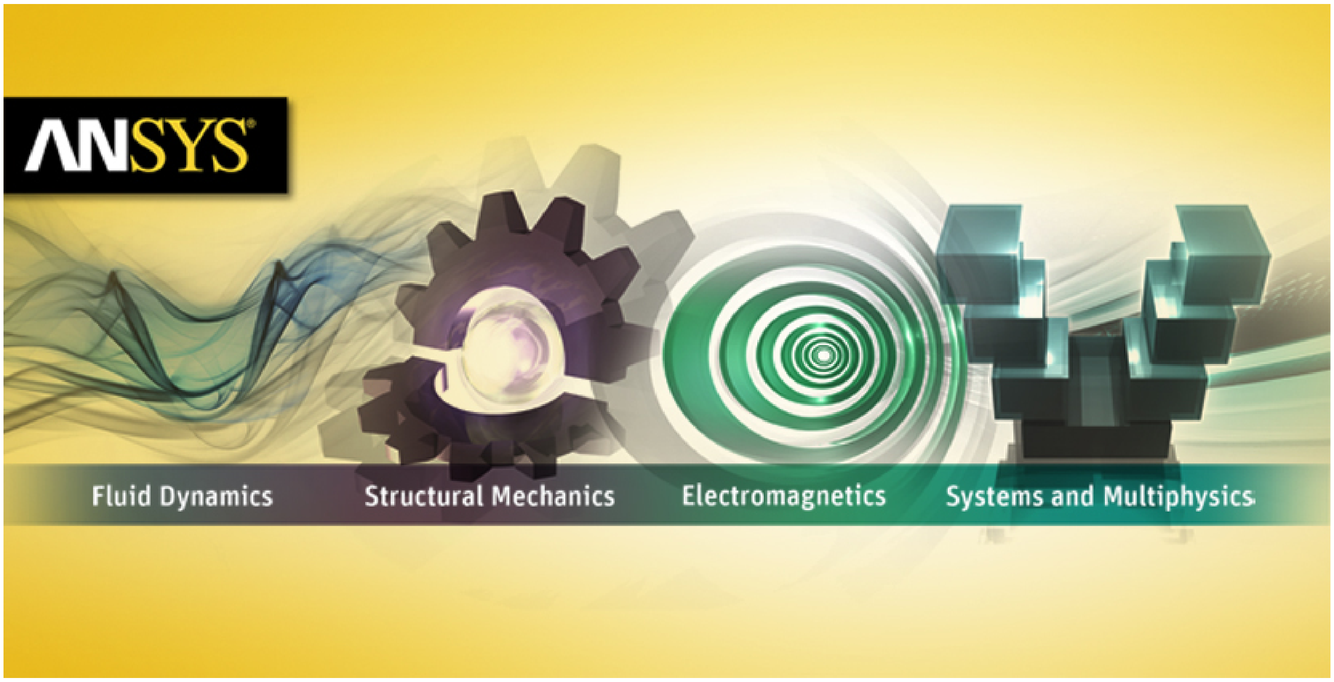


ANSYS FLUENT Population Balance Module Manual

ANSYS, Inc.
Southpointe
275 Technology Drive
Canonsburg, PA 15317
ansysinfo@ansys.com
http://www.ansys.com
(T) 724-746-3304
(F) 724-514-9494

Release 14.0
November 2011

Copyright and Trademark Information

Disclaimer Notice

THIS ANSYS SOFTWARE PRODUCT AND PROGRAM DOCUMENTATION INCLUDE TRADE SECRETS AND ARE CONFIDENTIAL AND PROPRIETARY PRODUCTS OF ANSYS, INC., ITS SUBSIDIARIES, OR LICENSORS. The software products and documentation are furnished by ANSYS, Inc., its subsidiaries, or affiliates under a software license agreement that contains provisions concerning non-disclosure, copying, length and nature of use, compliance with exporting laws, warranties, disclaimers, limitations of liability, and remedies, and other provisions. The software products and documentation may be used, disclosed, transferred, or copied only in accordance with the terms and conditions of that software license agreement.

ANSYS, Inc. is certified to ISO 9001:2008.

U.S. Government Rights

For U.S. Government users, except as specifically granted by the ANSYS, Inc. software license agreement, the use, duplication, or disclosure by the United States Government is subject to restrictions stated in the ANSYS, Inc. software license agreement and FAR 12.212 (for non-DOD licenses).

Third-Party Software

See the legal information in the product help files for the complete Legal Notice for ANSYS proprietary software and third-party software. If you are unable to access the Legal Notice, please contact ANSYS, Inc.

Published in the U.S.A.

Table of Contents

Using This Manual

This preface is divided into the following sections:

1. 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.

2. The Contents of the FLUENT Manuals

The manuals listed below form the FLUENT product documentation set. They include descriptions of the procedures, commands, and theoretical details needed to use FLUENT products.

FLUENT Getting Started Guide contains general information about getting started with using FLUENT.
FLUENT User's Guide contains detailed information about using FLUENT, including information about the user interface, reading and writing files, defining boundary conditions, setting up physical models, calculating a solution, and analyzing your results.
FLUENT in Workbench User's Guide contains information about getting started with and using FLUENT within the Workbench environment.
FLUENT Theory Guide contains reference information for how the physical models are implemented in FLUENT.
FLUENT UDF Manual contains information about writing and using user-defined functions (UDFs).
FLUENT Tutorial Guide contains a number of example problems with detailed instructions, commentary, and postprocessing of results.
FLUENT Text Command List contains a brief description of each of the commands in FLUENT's text interface.
FLUENT Adjoint Solver Module Manual contains information about the background and usage of FLUENT's Adjoint Solver Module that allows you to obtain detailed sensitivity data for the performance of a fluid system.
FLUENT Battery Module Manual contains information about the background and usage of FLUENT's Battery Module that allows you to analyze the behavior of electric batteries.
FLUENT Continuous Fiber Module Manual contains information about the background and usage of FLUENT's Continuous Fiber Module that allows you to analyze the behavior of fiber flow, fiber properties, and coupling between fibers and the surrounding fluid due to the strong interaction that exists between the fibers and the surrounding gas.
FLUENT Fuel Cell Modules Manual contains information about the background and the usage of two separate add-on fuel cell models for FLUENT that allow you to model polymer electrolyte membrane fuel cells (PEMFC), solid oxide fuel cells (SOFC), and electrolysis with FLUENT.
FLUENT Magnetohydrodynamics (MHD) Module Manual contains information about the background and usage of FLUENT's Magnetohydrodynamics (MHD) Module that allows you to analyze

the behavior of electrically conducting fluid flow under the influence of constant (DC) or oscillating (AC) electromagnetic fields.

FLUENT Migration Manual contains information about transitioning from the previous release of FLUENT, including details about new features, solution changes, and text command list changes.

FLUENT Population Balance Module Manual contains information about the background and usage of FLUENT's Population Balance Module that allows you to analyze multiphase flows involving size distributions where particle population (as well as momentum, mass, and energy) require a balance equation.

Running FLUENT Under LSF contains information about the using FLUENT with Platform Computing's LSF software, a distributed computing resource management tool.

Running FLUENT Under PBS Professional contains information about the using FLUENT with Altair PBS Professional, an open workload management tool for local and distributed environments.

Running FLUENT Under SGE contains information about the using FLUENT with Sun Grid Engine (SGE) software, a distributed computing resource management tool.

3. Typographical Conventions

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

- Different type styles are used to indicate graphical user interface menu items and text interface menu items (for example, **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 → **Multiphase** → **Edit...**

  indicates that **Models** is selected in the navigation pane, which then opens the corresponding task page. In the **Models** task page, **Multiphase** is selected from the list. Clicking the **Edit...** button opens the **Multiphase** 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,

  Define → **Injections...**

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

  `display` → `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.

4. 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, I).
- The operator ∇, 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, ∇ is defined to be

$$\frac{\partial}{\partial x}\vec{\imath}+\frac{\partial}{\partial y}\vec{\jmath}+\frac{\partial}{\partial z}\vec{k} \tag{1}$$

  ∇ appears in several ways:

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

$$\nabla p=\frac{\partial p}{\partial x}\vec{\imath}+\frac{\partial p}{\partial y}\vec{\jmath}+\frac{\partial p}{\partial z}\vec{k} \tag{2}$$

  - The gradient of a vector quantity is a second-order tensor; for example, in Cartesian coordinates,

$$\nabla(\vec{v})=\left(\frac{\partial}{\partial x}\vec{\imath}+\frac{\partial}{\partial y}\vec{\jmath}+\frac{\partial}{\partial z}\vec{k}\right)\left(v_x\vec{\imath}+v_j\vec{\jmath}+v_z\vec{k}\right) \tag{3}$$

    This tensor is usually written as

$$\begin{pmatrix} \frac{\partial v_x}{\partial x} & \frac{\partial v_x}{\partial y} & \frac{\partial v_x}{\partial z} \\ \frac{\partial v_y}{\partial x} & \frac{\partial v_y}{\partial y} & \frac{\partial v_y}{\partial z} \\ \frac{\partial v_z}{\partial x} & \frac{\partial v_z}{\partial y} & \frac{\partial v_z}{\partial z} \end{pmatrix} \tag{4}$$

  - The divergence of a vector quantity, which is the inner product between ∇ and a vector; for example,

$$\nabla \cdot \vec{v} = \frac{\partial v_x}{\partial x} + \frac{\partial v_y}{\partial y} + \frac{\partial v_z}{\partial z} \qquad \textbf{(5)}$$

- The operator $\nabla \cdot \nabla$, which is usually written as ∇^2 and is known as the Laplacian; for example,

$$\nabla^2 T = \frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} + \frac{\partial^2 T}{\partial z^2} \qquad \textbf{(6)}$$

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

$$(\nabla T)^2 = \left(\frac{\partial T}{\partial x}\right)^2 + \left(\frac{\partial T}{\partial y}\right)^2 + \left(\frac{\partial T}{\partial z}\right)^2 \qquad \textbf{(7)}$$

- An exception to the use of ∇ is found in the discussion of Reynolds stresses in "Modeling Turbulence" in the User's Guide, where convention dictates the use of Cartesian tensor notation. In this chapter, you will also find that some velocity vector components are written as u, v, and w instead of the conventional v with directional subscripts.

5. 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 ANSYS Customer Portal (www.ansys.com/customerportal).

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.

Technical Support for ANSYS, Inc. products is provided either by ANSYS, Inc. directly or by one of our certified ANSYS Support Providers. Please check with the ANSYS Support Coordinator (ASC) at your company to determine who provides support for your company, or go to www.ansys.com and select **About ANSYS> Contacts and Locations**. The direct URL is: http://www1.ansys.com/customer/public/supportlist.asp. Follow the on-screen instructions to obtain your support provider contact information. You will need your customer number. If you don't know your customer number, contact the ASC at your company.

If your support is provided by ANSYS, Inc. directly, Technical Support can be accessed quickly and efficiently from the ANSYS Customer Portal, which is available from the ANSYS Website (www.ansys.com) under **Support> Technical Support** where the Customer Portal is located. The direct URL is: http://www.ansys.com/customerportal.

One of the many useful features of the Customer Portal is the Knowledge Resources Search, which can be found on the Home page of the Customer Portal.

Systems and installation Knowledge Resources are easily accessible via the Customer Portal by using the following keywords in the search box: `Systems/Installation`. These Knowledge Resources provide solutions and guidance on how to resolve installation and licensing issues quickly.

NORTH AMERICA

All ANSYS, Inc. Products

Web: Go to the ANSYS Customer Portal (http://www.ansys.com/customerportal) and select the appropriate option.

Toll-Free Telephone: 1.800.711.7199

Fax: 1.724.514.5096

Support for University customers is provided only through the ANSYS Customer Portal.

GERMANY

ANSYS Mechanical Products

Telephone: +49 (0) 8092 7005-55

Email: support@cadfem.de

All ANSYS Products

Web: Go to the ANSYS Customer Portal (http://www.ansys.com/customerportal) and select the appropriate option.

National Toll-Free Telephone:

German language: 0800 181 8499

English language: 0800 181 1565

International Telephone:

German language: +49 6151 3644 300

English language: +49 6151 3644 400

Email: support-germany@ansys.com

UNITED KINGDOM

All ANSYS, Inc. Products

Web: Go to the ANSYS Customer Portal (http://www.ansys.com/customerportal) and select the appropriate option.

Telephone: +44 (0) 870 142 0300

Fax: +44 (0) 870 142 0302

Email: support-uk@ansys.com

Support for University customers is provided only through the ANSYS Customer Portal.

JAPAN

CFX , ICEM CFD and Mechanical Products

Telephone: +81-3-5324-8333

Fax: +81-3-5324-7308

Email: *CFX:* japan-cfx-support@ansys.com; *Mechanical:* japan-ansys-support@ansys.com

FLUENT Products

Telephone: +81-3-5324-7305

Email: *FLUENT:* japan-fluent-support@ansys.com; *POLYFLOW:* japan-polyflow-support@ansys.com; *FfC:* japan-ffc-support@ansys.com; *FloWizard:* japan-flowizard-support@ansys.com

Icepak

Telephone: +81-3-5324-7444

Email: japan-icepak-support@ansys.com

Licensing and Installation

Email: japan-license-support@ansys.com

INDIA

ANSYS Products (including FLUENT, CFX, ICEM-CFD)

Web: Go to the ANSYS Customer Portal (http://www.ansys.com/customerportal) and select the appropriate option.

Telephone: +91 1 800 233 3475 (toll free) or +91 1 800 209 3475 (toll free)

Fax: +91 80 2529 1271

Email: *FEA products:* feasup-india@ansys.com; *CFD products:* cfdsup-india@ansys.com; *Installation:* installation-india@ansys.com

FRANCE

All ANSYS, Inc. Products

Web: Go to the ANSYS Customer Portal (http://www.ansys.com/customerportal) and select the appropriate option.

Toll-Free Telephone: +33 (0) 800 919 225

Email: support-france@ansys.com

Support for University customers is provided only through the ANSYS Customer Portal.

BELGIUM

All ANSYS Products

Web: Go to the ANSYS Customer Portal (http://www.ansys.com/customerportal) and select the appropriate option.

Telephone: +32 (0) 10 45 28 61

Email: support-belgium@ansys.com

Support for University customers is provided only through the ANSYS Customer Portal.

SWEDEN

All ANSYS Products

Web: Go to the ANSYS Customer Portal (http://www.ansys.com/customerportal) and select the appropriate option.

Telephone: +44 (0) 870 142 0300

Email: support-sweden@ansys.com

Support for University customers is provided only through the ANSYS Customer Portal.

SPAIN and PORTUGAL

All ANSYS Products

Web: Go to the ANSYS Customer Portal (http://www.ansys.com/customerportal) and select the appropriate option.

Telephone: +33 1 30 60 15 63

Email: support-spain@ansys.com

Support for University customers is provided only through the ANSYS Customer Portal.

ITALY

All ANSYS Products

Web: Go to the ANSYS Customer Portal (http://www.ansys.com/customerportal) and select the appropriate option.

Telephone: +39 02 89013378

Email: support-italy@ansys.com

Support for University customers is provided only through the ANSYS Customer Portal.

Chapter 1: Introduction

In ANSYS FLUENT the population balance model is provided as an add-on 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 (for example, particle size, porosity, composition, and so on), 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.

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 Inhomogeneous Discrete Method

One of the limitations of the existing homogeneous discrete method is that all bins are assigned to the same secondary phase and are therefore advected with the same phase momentum. This is unsuitable for modeling cases where large and small bin sizes are likely to segregate due to different momentum fields. The inhomogeneous discrete method overcomes this limitation by allowing groups of bins to be advected by different phase velocities. Thus when the inhomogeneous discrete model is activated, the Population Balance model can be applied to more than one secondary phase.

The general transport equation for the discrete bin fraction f_i can be written as

$$\frac{\partial}{\partial t}(\rho \alpha f_i) + \nabla \cdot (\overrightarrow{u_p} \alpha f_i) = S_{bi} \tag{1–1}$$

Since all bins belong to a single phase in the homogeneous discrete method, the net mass source for the phase in case of breakage and agglomeration is zero and can be expressed as

$$\sum_{i=1}^{M} S_{bi} = 0 \tag{1–2}$$

This is shown schematically in *Figure 1.1* (p. 3) where all bins are advected by the same phase velocity u_p. In contrast, the inhomogeneous discrete method shown in *Figure 1.2* (p. 3) allows bins to be assigned to multiple phases. Here M bins per phase are distributed over N phases for a total of $M \times N$ bins. Bins f_1 and f_M are advected by phase velocity u_{p_1} and so forth. The sum of bin sources in any given phase is not necessarily equal to zero since bins in a given phase can migrate to another phase through breakage or agglomeration, thus creating a net mass source for that phase.

The net mass source for a given phase can be expressed as the sum of the bin sources belonging to that phase

$$S_i = \sum_{i=1}^{M} S_{bi} \tag{1–3}$$

For breakage and coalescence the sum over all phase sources is zero

$$\sum_{i=1}^{N} S_i = 0 \tag{1–4}$$

also similar to the homogeneous discrete model

$$\sum_{i=1}^{M} f_i = 1 \tag{1–5}$$

Important

The inhomogeneous discrete method is currently limited to breakage and coalescence only.

Figure 1.1 Homogeneous Discrete Method

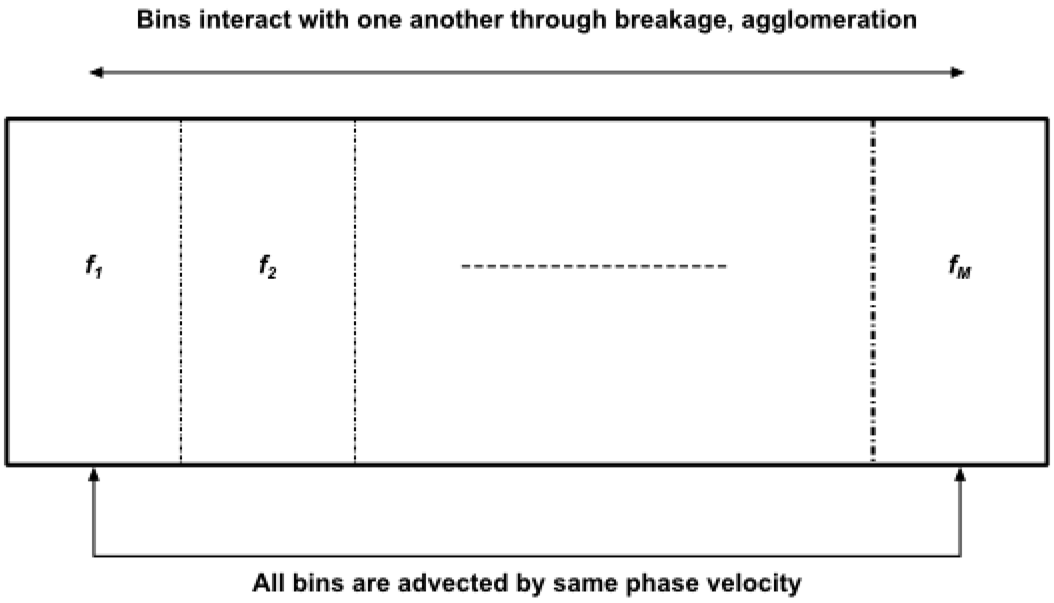
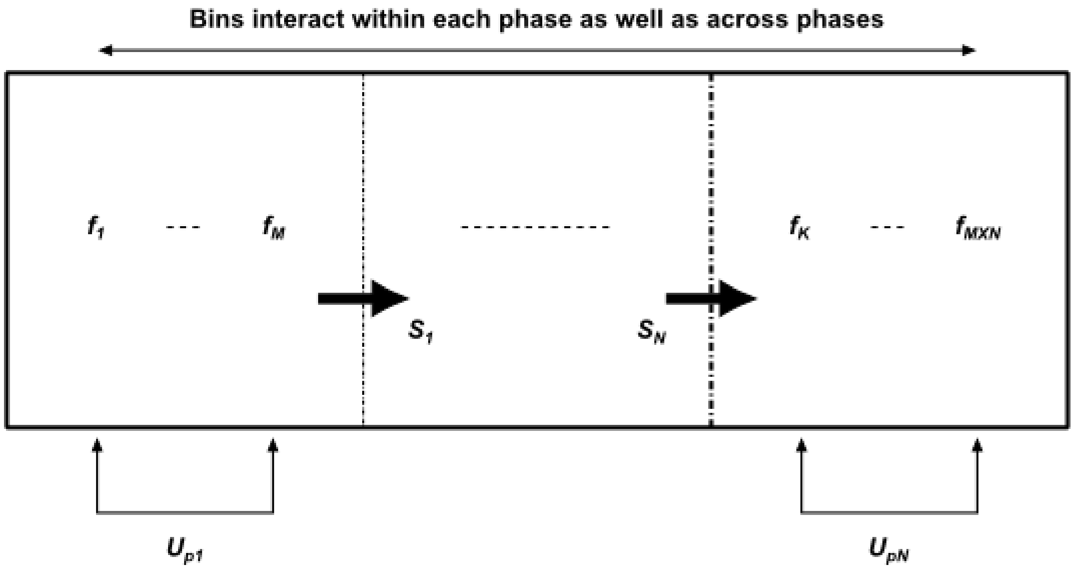

Bins interact with one another through breakage, agglomeration
f_1
f_2
f_M
All bins are advected by same phase velocity

Figure 1.2 Inhomogeneous Discrete Method



1.3. 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 ith moment is defined by integrating the number density throughout the particle space weighted with the particle property raised to its ith 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.

1.4. 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.

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" (ϕ), 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 $dV_{\vec{x}} dV_\phi$ is $n(\vec{x}, \phi, t)\, dV_{\vec{x}} dV_\phi$. 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

$$\int_{\Omega_\phi} \int_{\Omega_{\vec{x}}} n\, dV_{\vec{x}}\, dV_\phi \qquad \textbf{(2–1)}$$

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

$$N(\vec{x}, t) = \int_{\Omega_\phi} n\, dV_\phi \qquad \textbf{(2–2)}$$

The total volume fraction of all particles is given by

$$\alpha(\vec{x}, t) = \int_{\Omega_\phi} n V(\phi)\, dV_\phi \qquad \textbf{(2–3)}$$

where $V(\phi)$ is the volume of a particle in state ϕ.

2.2. The Population Balance Equation (PBE)

Assuming that ϕ is the particle volume, the transport equation for the number density function is given as

$$\frac{\partial}{\partial t}\left[n(V,t)\right] + \nabla \cdot \left[\vec{u}\, n(V,t)\right] + \underbrace{\nabla_V \cdot \left[G_V n(V,t)\right]}_{\text{Growth term}} = \tag{2–4}$$

$$\underbrace{\frac{1}{2}\int_0^V a(V-V',V')\, n(V-V',t)\, n(V',t)\, dV'}_{\text{Birth due to Aggregation}}$$

$$-\underbrace{\int_0^\infty a(V,V')\, n(V,t)\, n(V',t)\, dV'}_{\text{Death due to Aggregation}}$$

$$+\underbrace{\int_{\Omega_V} pg(V')\, \beta(V\,|\,V')\, n(V',t)\, dV'}_{\text{Birth due to Breakage}}$$

$$-\underbrace{g(V)\, n(V,t)}_{\text{Death due to Breakage}}$$

The boundary and initial conditions are given by

$$n(V,t=0) = n_V;\; n(V=0,t)\, G_V = \dot{n}_0 \tag{2–5}$$

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

2.2.1. Particle Growth and Dissolution

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

$$G_V = \frac{\partial V}{\partial t} \tag{2–6}$$

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

$$G = \frac{\partial L}{\partial t} \qquad \textbf{(2–7)}$$

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

$$G_v = 3 K_v L^2 G \qquad \textbf{(2–8)}$$

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

Important

Dissolution of particles can be represented as negative growth.

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.

2.2.2.1. Breakage

The breakage rate expression, or kernel [*18*] (p. 66), is expressed as

$$g\,(V')\,\beta\,(V \mid V')$$

where

$g\,(V')$ = breakage frequency; that is, the fraction of particles of volume V' breaking per unit time $(\mathrm{m}^{-3}\mathrm{s}^{-1})$

$\beta\,(V \mid V')$ = 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

$$B_{br} = \int_{\Omega_v} pg\,(V')\,\beta\,(V \mid V')\,n\,(V')\,dV' \qquad \textbf{(2–9)}$$

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

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

$$D_{br} = g(V)\, n(V) \tag{2–10}$$

The PDF $\beta(V \mid V')$ 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:

$$\int_0^{V'} \beta(V \mid V')\, dV = 1 \tag{2–11}$$

- For conservation of mass

$$p \int_0^{V'} m(V)\, \beta(V \mid V')\, dV = m(V') \tag{2–12}$$

- For binary breakage, β is symmetric about $V/V' = 0.5$; that is,

$$\beta(V' - V \mid V') = \beta(V \mid V') \tag{2–13}$$

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

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

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

- parabolic PDF
- Laakkonen 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.

2.2.2.2. 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 [15] (p. 65) as

$$\Omega_{br}(V,V') = \Omega_B(V')\ \eta(V\mid V')\ [1/m^3/sec] \tag{2–14}$$

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 λ 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

$$\Omega_{br}(V,V') = K\int_{\xi_{min}}^{1}\frac{(1+\xi)^2}{\xi^n}exp(-b\xi^m)\,d\xi \tag{2–15}$$

where the parameters are as shown in the tables that follow:

Table 2.1 Luo Model Parameters

$K\ [1/m^3/sec]$	n	b	m
$0.9238\varepsilon^{1/3}d^{-2/3}\alpha$	$11/3$	$12[f^{2/3}+(1-f)^{2/3}-1]\sigma\rho^{-1}\varepsilon^{-2/3}d^{-5/3}\beta^{-1}$	$-11/3$

Where β=2.047. See Luo [18] (p. 66).

Table 2.2 Lehr Model Parameters

$K\ [1/m^3/sec]$	n	b	m
$1.19\varepsilon^{-1/3}d^{-7/3}\sigma\rho^{-1}f^{-1/3}$	$13/3$	$2We_{crit}\sigma\rho^{-1}\varepsilon^{-2/3}d^{-5/3}f^{-1/3}$	$-2/3$

Where We_{crit} is input through the GUI. See Lehr[15] (p. 65).

2.2.2.3. Ghadiri Breakage Kernels

The Ghadiri model [7] (p. 65), [22] (p. 66), 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:

$$f = \frac{\rho_s E^{2/3}}{\Gamma^{5/3}} v^2 L^{5/3} = K_b v^2 L^{5/3} \tag{2–16}$$

where ρ_s is the particle density, E is the elastic modulus of the granule, and Γ 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

$$K_b = \frac{\rho_s E^2/3}{\Gamma^{5/3}} \tag{2–17}$$

2.2.2.4. Laakkonen Breakage Kernels

The Laakkonen breakage kernel is expressed as the product of a breakage frequency, $g(V')$ and a daughter PDF $\beta(V, V')$ where

$$g(V') = C_2 \varepsilon^{1/3} erfc\left(\sqrt{C_3 \frac{\sigma}{\rho_L \varepsilon^{2/3} d^{5/3}} + C_4 \frac{\mu_L}{\sqrt{\rho_L \rho_G} \varepsilon^{2/3} d^{5/3}}} \right) \tag{2–18}$$

where ε is the liquid phase eddy dissipation, σ is the surface tension, ρ_l is the liquid density, ρ_g is the gas density, d is the parent particle diameter and μ_L is the liquid viscosity.

The constants $C_2 = 2.52$, $C_3 = 0.04$ and $C_4 = 0.01$.

The daughter PDF is given by

$$\beta(V, V') = \frac{30}{V'} \left(\frac{V}{V'} \right)^2 \left(1 - \frac{V}{V'} \right)^2 \tag{2–19}$$

where V and V' are the daughter and parent particle volumes, respectively. This model is a useful alternative to the widely used Luo model because it has a simple expression for the daughter PDF and therefore requires significantly less computational effort.

2.2.2.5. 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 enables you to define the breakage PDF such that

$$\beta(V|V') = 0.5\left[\frac{C}{V'} + \frac{1-C/2}{V'}\left\{24\left(\frac{V}{V'}\right)^2 - 24\left(\frac{V}{V'}\right) + 6\right\}\right] \tag{2–20}$$

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 2–20* (p. 11) is symmetric about $V/V'=0.5$.

2.2.2.6. Generalized PDF

The generalized form of the PDF implemented in ANSYS FLUENT enables you to simulate multiple breakage fragments (>2) and to specify the form of the daughter distribution (for example, 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*] (p. 66) where the similarity z is the ratio of daughter-to-parent size (that is, $z \equiv \frac{V}{V'}$), then the generalized PDF is given by

$$p\beta(V|V') = \frac{\theta(z)}{V'} \tag{2–21}$$

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

$$b_k = \int_0^1 z^k \theta(z)\,dz = \frac{B_k(V')}{V'^k} \tag{2–22}$$

where

$$B_k(V') = \int_0^{V'} V^k p\beta(V|V')\,dV \tag{2–23}$$

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

$$b_0 = \int_0^1 \theta(z)\, dz = p \tag{2–24}$$

$$b_1 = \int_0^1 z\theta(z)\, dz = 1 \tag{2–25}$$

The generalized form of $\theta(z)$ [6] (p. 65) can be expressed as

$$\theta(z) = \sum_i w_i p_i \frac{z^{q_i - 1}(1-z)^{r_i - 1}}{\beta\left(q_i, r_i\right)} \tag{2–26}$$

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 $\beta\left(q_i, r_i\right)$ is the beta function. The following constraints are imposed on the parameters in *Equation 2–26* (p. 12) :

$$\sum_i w_i = 1 \tag{2–27}$$

$$\sum_i w_i p_i = p \tag{2–28}$$

$$\sum_i w_i \left(\frac{p_i q_i}{q_i + r_i} \right) = 1 \tag{2–29}$$

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

Table 2.3 Daughter Distributions

Type	$\theta(z)$
Equalized [12]	$p\delta\left(z - \frac{1}{p}\right)$
Attrition [12]	$\delta(z - (1-\varepsilon)) + \delta(z - \varepsilon)$
Power Law [29]	$(v+1)\, z^{v-1}$
Parabolic -a [29]	$(v+2)\,(v+1)\, z^{v-1}(1-z)$

Type	$\theta(z)$
Austin [2]	$w(v_1+1)\, z^{v_1-1} + (1-w)\,(v_2+1)\, z^{v_2-1}$
Binary Beta -a [11]	$60z^2(1-z)^2$
Binary Beta -b [20]	$\frac{2}{\beta(v,v)} z^{v-1}(1-z)^{v-1}$
Uniform [29]	$p(p-1)\,(1-z)^{p-2}$

Table 2.4 Daughter Distributions (cont.)

Type	p	Constraints
Equalized [12]	p	$p \geq 2$
Attrition [12]	2	$\varepsilon \ll 1$
Power Law [29]	$\frac{v+1}{v}$	$0 < v \leq 1$
Parabolic -a [29]	$\frac{v+2}{v}$	$0 < v \leq 2$
Austin [2]	$w\left(1+\frac{1}{v_1}\right) + (1-w)\left(1+\frac{1}{v_2}\right)$	$v_1, v_2 > 0; 1 \geq w \geq v_1\left(\frac{v_2-1}{v_2-v_1}\right)$
Binary Beta -a [11]	2	N/A
Binary Beta -b [20]	2	$v0$
Uniform [29]	p	$p \geq 2$

In *Table 2.3: Daughter Distributions* (p. 12), δ is the Dirac delta function, w is a weighting coefficient, and ε, v, v_1, and v_2 are user-defined parameters.

The generalized form can represent the daughter distributions in *Table 2.3: Daughter Distributions* (p. 12) by using the values shown in *Table 2.5: Values for Daughter Distributions in General Form* (p. 13).

Table 2.5 Values for Daughter Distributions in General Form

Type	$w0$	$p0$	$q0$	$r0$	$w1$	$p1$	$q1$	$r1$	Constraints
Equalized	1	p	$\infty *$	$\infty *$	N/A	N/A	N/A	N/A	$p \geq 2$
Attrition	0.5	2	ε	1	0.5	2	1	ε	$\varepsilon \ll 1$
Power Law	1	$\frac{v+1}{v}$	v	1	N/A	N/A	N/A	N/A	$0 < v \leq 1$
Parabolic	1	$\frac{v+2}{v}$	v	2	N/A	N/A	N/A	N/A	$0 < v \leq 2$
Austin	w	$\frac{v_1+1}{v_1}$	v_1	1	$1-w$	$\frac{v_2+1}{v_2}$	v_2	1	$v_1, v_2 > 0;$

Type	$w0$	$p0$	$q0$	$r0$	$w1$	$p1$	$q1$	$r1$	Constraints
									$1 \geq w \geq v_1 \left(\frac{v_2 - 1}{v_2 - v_1} \right)$
Binary Beta **	1	2	v	v	N/A	N/A	N/A	N/A	$v > 0$
Uniform	1	p	1	$p-1$	N/A	N/A	N/A	N/A	$p \geq 2$

(*)You can approximate ∞ by using a very large number, such as 1e20.

(**)Binary Beta -a is a special case of Binary Beta -b when $v = 3$.

Important

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.

2.2.2.7. Aggregation

The aggregation kernel [18] (p. 66) 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" (that is, 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

$$B_{ag} = \frac{1}{2} \int_0^V a\,(V - V', V')\, n\,(V - V')\, n\,(V')\, dV' \tag{2–30}$$

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

$$D_{ag} = \int_0^\infty a(V, V')\, n(V)\, n(V')\, dV' \tag{2–31}$$

Important

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.

2.2.2.8. Luo Aggregation Kernel

For the Luo model [17] (p. 65), 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:

$$\Omega_{ag}\left(V_i, V_j\right) = \omega_{ag}\left(V_i, V_j\right) P_{ag}\left(V_i, V_j\right) \ [m^3/sec] \tag{2–32}$$

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

$$\omega_{ag}\left(V_i, V_j\right) = \frac{\pi}{4}\left(d_i^2 + d_j^2\right) n_i n_j \overline{u}_{ij} \tag{2–33}$$

where $\overline{u}_{ij}$ is the characteristic velocity of collision of two particles with diameters d_i and d_j and number densities n_i and n_j.

$$\overline{u}_{ij} = \left(\overline{u}_i^2 + \overline{u}_j^2\right)^{1/2} \tag{2–34}$$

where

$$\bar{u}_i = 1.43\,(\varepsilon d_i)^{1/3} \tag{2–35}$$

The expression for the probability of aggregation is

$$P_{ag} = exp\left\{ -c_l \frac{\left[0.75\left(1+x_{ij}^2\right)\left(1+x_{ij}^3\right)\right]^{1/2}}{\left(\rho_2/\rho_1+0.5\right)^{1/2}\left(1+x_{ij}\right)^3} We_{ij}^{1/2} \right\} \tag{2–36}$$

where c_l is a constant of order unity, $x_{ij} = d_i/d_j$, ρ_1 and ρ_2 are the densities of the primary and secondary phases, respectively, and the Weber number is defined as

$$We_{ij} = \frac{\rho_l d_i\left(\bar{u}_{ij}\right)^2}{\sigma} \tag{2–37}$$

2.2.2.9. Free Molecular Aggregation Kernel

Real particles aggregate and break with frequencies (or kernels) characterized by complex dependencies over particle internal coordinates [28] (p. 66). 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:

$$a\left(L_i, L_j\right) = \frac{2k_B T}{3\mu}\frac{\left(L_i+L_j\right)^2}{L_i L_j} \tag{2–38}$$

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

2.2.2.10. 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, η, which is expressed as a function of the kinematic viscosity and the turbulent energy dissipation rate:

$$\eta = \left(\frac{v^3}{\varepsilon}\right)^{1/4} \tag{2–39}$$

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, v

- 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] (p. 66), the collision rate is expressed as,

$$a\left(L_i,L_j\right)=\varsigma_T\sqrt{\frac{8\pi}{15}}\dot{\gamma}\frac{\left(L_i+L_j\right)^3}{8} \tag{2–40}$$

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

$$\dot{\gamma}=\frac{\varepsilon^{0.5}}{v} \tag{2–41}$$

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] (p. 65),

$$a\left(L_i,L_j\right)=\varsigma_T 2^{3/2}\sqrt{\pi}\frac{\left(L_i+L_j\right)^2}{4}\sqrt{\left(U_i^2+U_j^2\right)} \tag{2–42}$$

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] (p. 65) proposed the following relationship:

$$\varsigma_T=0.732\left(\frac{5}{N_T}\right)^{0.242} \quad ; \quad N_T\geq 5 \tag{2–43}$$

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

$$N_T=\frac{6\pi\mu\left(L_i+L_j\right)^3\dot{\lambda}}{8H} \tag{2–44}$$

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

$$\dot{\lambda} = \left(\frac{4\varepsilon}{15\pi v}\right)^{0.5} \tag{2–45}$$

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 2–5* (p. 6).

2.3. Solution Methods

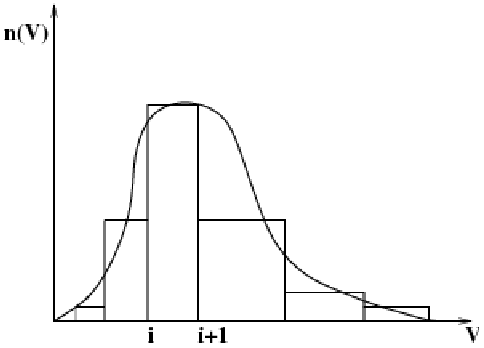
As discussed in *Introduction* (p. 1), the population balance equation can be solved by the four different methods in ANSYS FLUENT: the discrete method, the inhomogeneous discrete method, the standard method of moments (SMM), and the quadrature method of moments (QMOM). For each method, the ANSYS FLUENT 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 and the Inhomogeneous Discrete Method

The discrete method (also known as the classes or sectional method) was developed by Hounslow [*10*] (p. 65), Litster [*16*] (p. 65), and Ramkrishna [*25*] (p. 66). 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 2.1* (p. 18). 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 *a priori* and that a large number of classes may be required.

The solution methods for the inhomogeneous discrete method are based on the discrete method and therefore share many of the same fundamentals.

Figure 2.1 A Particle Size Distribution as Represented by the Discrete Method



2.3.1.1. Numerical Method

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

$$\frac{\partial}{\partial t}\left(\rho_s\alpha_i\right) + \nabla\cdot\left(\rho_s u_i\alpha_i\right) + \frac{\partial}{\partial V}\left(\frac{G_v\rho_s\alpha_i}{V}\right) = \tag{2–46}$$

$$\rho_s V_i\left(B_{ag,i} - D_{ag,i} + B_{br,i} - D_{br,i}\right) + 0^i\rho_s V_0\dot{n}_0$$

where ρ_s is the density of the secondary phase and α_i is the volume fraction of particle size i, defined as

$$\alpha_i = N_i V_i \qquad i = 0, 1, \cdots, N-1 \tag{2–47}$$

where

$$N_i(t) = \int_{V_i}^{V_{i+1}} n(V,t)\, dV \tag{2–48}$$

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

$$f_i = \frac{\alpha_i}{\alpha} \tag{2–49}$$

where α 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 0^i signifies that this particular term, in this case $\rho_s V_0\dot{n}_0$, appears in *Equation 2–46* (p. 19) only in the case of the smallest particle size.

The growth rate in is discretized as follows *Equation 2–46* (p. 19) [*10*] (p. 65):

$$\frac{\partial}{\partial V}\left(\frac{G_v\rho_s\alpha_i}{V}\right) = \rho_s V_i\left[\left(\frac{G_{v,i-1}N_{i-1}}{V_i - V_{i-1}}\right) - \left(\frac{G_{v,i}N_i}{V_{i+1} - V_i}\right)\right] \tag{2–50}$$

The volume coordinate is discretized as [*10*] (p. 65) $V_{i+1}/V_i = 2^q$ where $q = 1, 2, \ldots$ and is referred to as the "ratio factor".

The particle birth and death rates are defined as follows:

$$B_{ag,i} = \sum_{k=1}^{N}\sum_{j=1}^{N} a_{kj} N_k N_j x_{kj} \xi_{kj} \tag{2–51}$$

$$D_{ag,i} = \sum_{j=1}^{N} a_{ij} N_i N_j \tag{2–52}$$

$$B_{br,i} = \sum_{j=i+1}^{N} g\left(V_j\right) N_j \beta\left(V_i \mid V_j\right) \tag{2–53}$$

$$D_{br,i} = g\left(V_i\right) N_i \tag{2–54}$$

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

$$\xi_{kj} = \begin{cases} 1 & \text{for } V_i < V_{ag} < V_{i+1}, \text{ where } i \le N-1 \\ 0 & \text{otherwise} \end{cases} \tag{2–55}$$

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

$$V_{ag} = \left[x_{kj} V_i + \left(1 - x_{kj}\right) V_{i+1}\right] \tag{2–56}$$

where

$$x_{kj} = \frac{V_{ag} - V_{i+1}}{V_i - V_{i+1}} \tag{2–57}$$

If V_{ag} is greater than or equal to the largest particle size V_N, then the contribution to class $N-1$ is

$$x_{kj} = \frac{V_{ag}}{V_N} \tag{2–58}$$

Important

Note that there is no breakage for the smallest particle class.

2.3.1.2. Breakage Formulations for the Discrete Method

The default breakage formulation for the discrete method in ANSYS FLUENT is based on the Hagesather method [14] (p. 65). 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

$$B_b(i) = \sum_{k=i+1, i \neq N}^{N} \Omega_b(v_k, v_i) + \sum_{k=i, i \neq N}^{i} x_{i+1,k} \Omega_b(v_{i+1}, v_k) + \tag{2–59}$$

$$\sum_{k=1, i \neq 1}^{i-1} \left(1 - x_{i,k}\right) g(v_{i+1} \Omega_b(v_i, v_k))$$

Here

$$\Omega_b(v_k, v_i) = N_k g(v_k) \beta(v_k, v_i) \tag{2–60}$$

A more mathematically rigorous formulation is given by Ramkrishna[13] (p. 65), where the breakage rate is expressed as

$$B_b(i) = \sum_{i}^{N} n_{i,k} g(v_k) N_k \tag{2–61}$$

where

$$n_{i,k} = \int_{v_i} v_{i+1} \frac{v_{i+1} - v}{v_{i+1} - v_i} \beta(v_k, v)\, dv + \int_{v_{i-1}}^{v_i} \frac{v - v_{i-1}}{v_i - vi - 1} \beta(v_k, v)\, dv \tag{2–62}$$

The Ramkrishna formulation can be slow due to the large number of integration points required. However, for simple forms of β, the integrations can be performed relatively easily. The Hagesather formulation requires fewer integration points and the difference in accuracy with the Ramkrishna formulation can be corrected by a suitable choice of bin sizes.

Important

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 Ramkrishna formulation.

Note

The inhomogeneous discrete phase applies the Hagesather formulation.

2.3.2. The Standard Method of Moments (SMM)

The SMM, proposed by Randolph and Larson [26] (p. 66) 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 *The Quadrature Method of Moments (QMOM)* (p. 24)).

2.3.2.1. 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 kth moment as

$$m_k(\vec{x},t) = \int_0^\infty n(L;\ \vec{x},t)\, L^k dL \qquad k = 0, 1, \cdots, N-1 \tag{2–63}$$

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

$$\frac{\partial}{\partial t}(\rho_s m_k) + \nabla \cdot (\rho \vec{u}\, m_k) = \rho_s(\overline{B}_{\mathrm{ag},k} - \overline{D}_{\mathrm{ag},k} + \overline{B}_{\mathrm{br},k} - \overline{D}_{\mathrm{br},k}) + 0^k \dot{n}_0 + \mathrm{Growth} \tag{2–64}$$

where

$$\overline{B}_{ag,k} = \frac{1}{2}\int_0^\infty n(\lambda) \int_0^\infty a(u,\lambda)\ (u,\lambda)\ (u^3+\lambda^3)^{k/3}\, n(u)\, du d\lambda \tag{2–65}$$

$$\overline{D}_{ag,k} = \int_0^\infty L^k n(L) \int_0^\infty a(L,\lambda)\, n(\lambda)\, d\lambda dL \tag{2–66}$$

$$\overline{B}_{br,k} = \int_0^\infty L^k \int_0^\infty g(\lambda)\, \beta(L \mid \lambda)\, n(\lambda)\, d\lambda dL \tag{2–67}$$

$$\overline{D}_{br,k} = \int_0^k L^k g(L)\, n(L)\, dL \tag{2–68}$$

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

$$\text{Growth} \equiv \int_0^\infty kL^{k-1} G(L)\, n(L,t)\, dL \tag{2–69}$$

and for constant growth is represented as

$$kGm_{k-1} \tag{2–70}$$

Equation 2–65 (p. 22) can be derived by using

$$u^3 = L^3 - \lambda^3;\ dL = \frac{u^2}{L^2} du$$

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

$$N_{total} = m_0 \tag{2–71}$$

$$L_{total} = m_1 \tag{2–72}$$

$$A_{total} = K_a m_2 \tag{2–73}$$

$$V_{total} = K_v m_3 \tag{2–74}$$

$$d_{32} = \frac{m_3}{m_2} \tag{2–75}$$

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 2–64 (p. 22), the quantities represented in Equation 2–65 (p. 22) – Equation 2–68 (p. 22) 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 Equation 2–65 (p. 22) – Equation 2–68 (p. 22) 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] (p. 66) for modeling aerosol evolution and coagulation problems. Its applications by Marchisio et al. [19] (p. 66) 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.

2.3.3.1. Numerical Method

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

$$\int_0^\infty f(L)\, n(L)\, dL \approx \sum_{i=1}^{N} f(L_i)\, w_i, \tag{2–76}$$

is exact if $f(L)$ is a polynomial of order N or smaller [5] (p. 65). 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:

$$m_k = \sum_{i=1}^{N} w_i L_i^k. \tag{2–77}$$

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] (p. 66). This procedure is known as the Product-Difference algorithm [8] (p. 65). 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 *Equation 2–76* (p. 24) and *Equation 2–77* (p. 24), the birth and death terms in *Equation 2–64* (p. 22) can be rewritten as

$$\overline{B}_{ag,k} = \frac{1}{2}\sum_{i=1}^{N} w_i \sum_{j=1}^{N} w_j \left(L_i^3 + L_j^3\right)^{k/3} a\left(L_i, L_j\right) \quad \text{(2–78)}$$

$$\overline{D}_{ag,k} = \sum_{i=1}^{N} L_i^k w_i \sum_{j=1}^{N} w_j a\left(L_i, L_j\right) \quad \text{(2–79)}$$

$$\overline{B}_{br,k} = \sum_{i=1}^{N} w_i \int_0^{\infty} L_k g\left(L_i\right) \beta\left(L \mid L_i\right) dL \quad \text{(2–80)}$$

$$\overline{D}_{br,k} = \sum_{i=1}^{N} w_i L_i^k g\left(L_i\right) \quad \text{(2–81)}$$

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 2–69* (p. 23) and represented as

$$\sum_{i=1}^{N} w_i L_i^{k-1} G\left(L_i\right) \quad \text{(2–82)}$$

to allow for a size-dependent growth rate.

2.3.4. The Direct Quadrature Method of Moments (DQMOM)

DQMOM equations are derived from the basic number density function equation via the moment transfer method, in a similar way to QMOM. The difference is that DQMOM assumes that each Quadrature point will occupy an independent velocity field, whereas QMOM assumes that all Quadrature points are moving on the same velocity field. This difference enables DQMOM to predict particle segregation due to particle interaction.

In this implementation of DQMOM, four phases must be specified: one primary phase and three secondary phases that are DQMOM phases. Compared to QMOM, for a three Quadrature points system, the DQMOM method only needs three extra equations to solve for the effective length of the particle, but there are additional source terms for the volume fraction equation for each DQMOM phase. In ANSYS FLUENT, three particle interactions are accounted for, growth, aggregation, and breakage. Nucleation is not considered.

2.3.4.1. Numerical Method

The DQMOM equations, describing a poly-dispersed particle system undergoing aggregation, breakage, and growth can be written as follows (the details of the DQMOM formulation can be found in [*30*] (p. 66)):

$$\frac{\partial \varepsilon_i \rho_s}{\partial t} + \nabla \cdot (\bar{u}_{si} \varepsilon_i \rho_s) = 3 k_v \rho_s L_i^2 (b_i + w_i G_i) - 2 k_v \rho_s L_i^3 a_i \tag{2–83}$$

$$\frac{\partial \varepsilon_i L_i \rho_s}{\partial t} + \nabla \cdot (\bar{u}_{si} \varepsilon_i L_i \rho_s) = 4 k_v \rho_s L_i^3 (b_i + w_i G_i) - 3 k_v \rho_s L_i^4 a_i \tag{2–84}$$

where ε_i and $\varepsilon_i L_i$ are the VOF and the effective length of the particle phase, respectively. w_i is the number of particles per unit volume and G_i is the growth rate at Quadrature point i, while a_i and b_i can be computed through a linear system resulting from the moment transformation of the particle number density transport equation using N Quadrature points. The linear system can be written in matrix form as

$$A\alpha = d \tag{2–85}$$

Where the $2N \times 2N$ coefficient matrix $A = [A_1 A_2]$ is defined by

$$A_1 = \begin{bmatrix} 1 & \dots & 1 \\ 0 & \dots & 0 \\ -L_1^2 & \dots & -L_N^2 \\ \vdots & \ddots & \vdots \\ 2(1-N)L_1^{2N-1} & \dots & 2(1-N)L_N^{2N-1} \end{bmatrix} \tag{2–86}$$

$$A_2 = \begin{bmatrix} 0 & \dots & 0 \\ 1 & \dots & 1 \\ 2L_1 & \dots & L_N \\ \vdots & \ddots & \vdots \\ (2N-1)L_1^{2N-2} & \dots & (2N-1)L_N^{2N-2} \end{bmatrix} \tag{2–87}$$

The $2\mathrm{N}$ vector of unknowns α is defined by

$$\alpha = [\, a_1 \ \dots \ a_N \ \ b_1 \ \dots \ b_N]^T = \begin{bmatrix} a \\ b \end{bmatrix} \tag{2–88}$$

The right hand side of *Equation 2–85* (p. 26) is the known source terms involving aggregation and breakage phenomena only. The growth term is accounted for directly in *Equation 2–83* (p. 26) and *Equation 2–84* (p. 26). At present, nucleation is not considered.

$$d = \left[\begin{array}{ccc} S_0^{(N)} & \dots & S_{2N-1}^{(N)} \end{array}\right]^T \tag{2–89}$$

The source term for the k^{th} moment $S_k^{(N)}$ $(k = 0, \dots, 2N-1)$ is defined as

$$S_k^{(N)}(x,t) = \int_0^{\infty} L^k S(x,t)\, dL \tag{2–90}$$

When the abscissas of the Quadrature points L_i are distinct, the matrix A is well defined and a unique solution of *Equation 2–85* (p. 26) can be obtained. Otherwise, the matrix A is not full rank and cannot be inverted to find a unique solution for α. The method adopted by ANSYS FLUENT to overcome this problem is to employ a perturbation technique. For example, for the current three Quadrature points system, the perturbation technique will add a small value to the abscissas to make sure the matrix A is full rank. It is important to note that the perturbation technique is only used for the definition of matrix A and no modifications are made to the source term vector of *Equation 2–90* (p. 27). Therefore, both the weights and overall source terms resulting from aggregation and breakage are not affected by the perturbation method. The simulation tests have found that the perturbation method can stabilize the solutions of *Equation 2–85* (p. 26) and reduce the physically unrealistically large source terms for the two phases whose abscissa are too close in value. However, the technique has little effect on the phase whose abscissa L_i is distinct from the other two.

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*] (p. 66), which was adapted for crystallization problems by Baldyga and Orciuch [*3*] (p. 65).

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

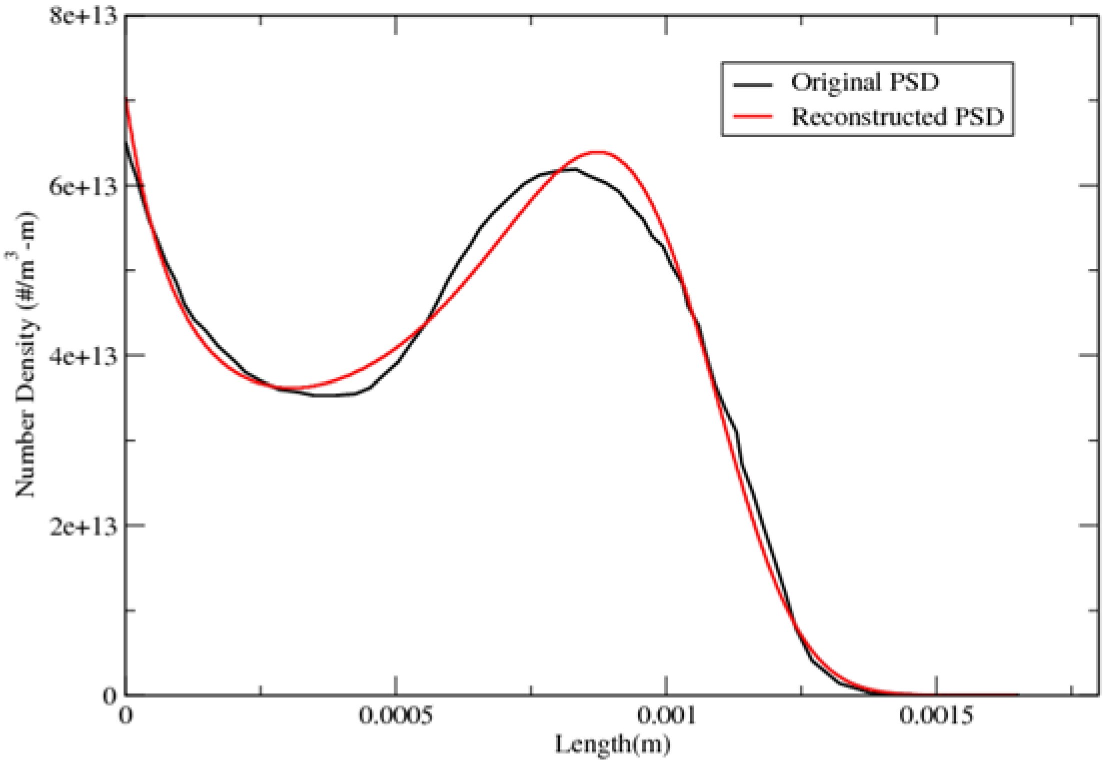
$$n(L) = \exp\left(\sum_{i=0}^{N-1} A_i L^i\right) \tag{2–91}$$

The equation for the kth moment is now written as

$$m_k = \int_0^{\infty} L^k \exp\left(\sum_{i=0}^{N-1} A_i L^i\right) dL \qquad k = 0, 1, \cdots, N-1 \tag{2–92}$$

Given N moments, the coefficients A_i can be found by a globally convergent Newton-Raphson method to reconstruct the particle size distribution (for example, *Figure 2.2* (p. 28)).

Figure 2.2 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:

3.1. Population Balance Module Installation

The population balance module is provided as an add-on 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

`define` → `models` → `addon-module`.

A list of ANSYS FLUENT add-on modules is displayed:

```
> /define/models/addon-module
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
     6. Adjoint Solver
     7. Battery Module
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 enable 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.

Important

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. However, if you are using the **Inhomogeneous Discrete**, more than one secondary phase can be used. Note that the properties of the secondary phases selected for that method should be the same for consistency.

For more information, please see the following sections:

3.3.1. Enabling the Population Balance Model
3.3.2. Defining Population Balance Boundary Conditions
3.3.3. Specifying Population Balance Solution Controls
3.3.4. Coupling With Fluid Dynamics
3.3.5. Specifying Interphase Mass Transfer Due to Nucleation and Growth

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, and so on, as usual. See the ANSYS FLUENT User's Guide for details.)

1. Start the double-precision version of ANSYS FLUENT.

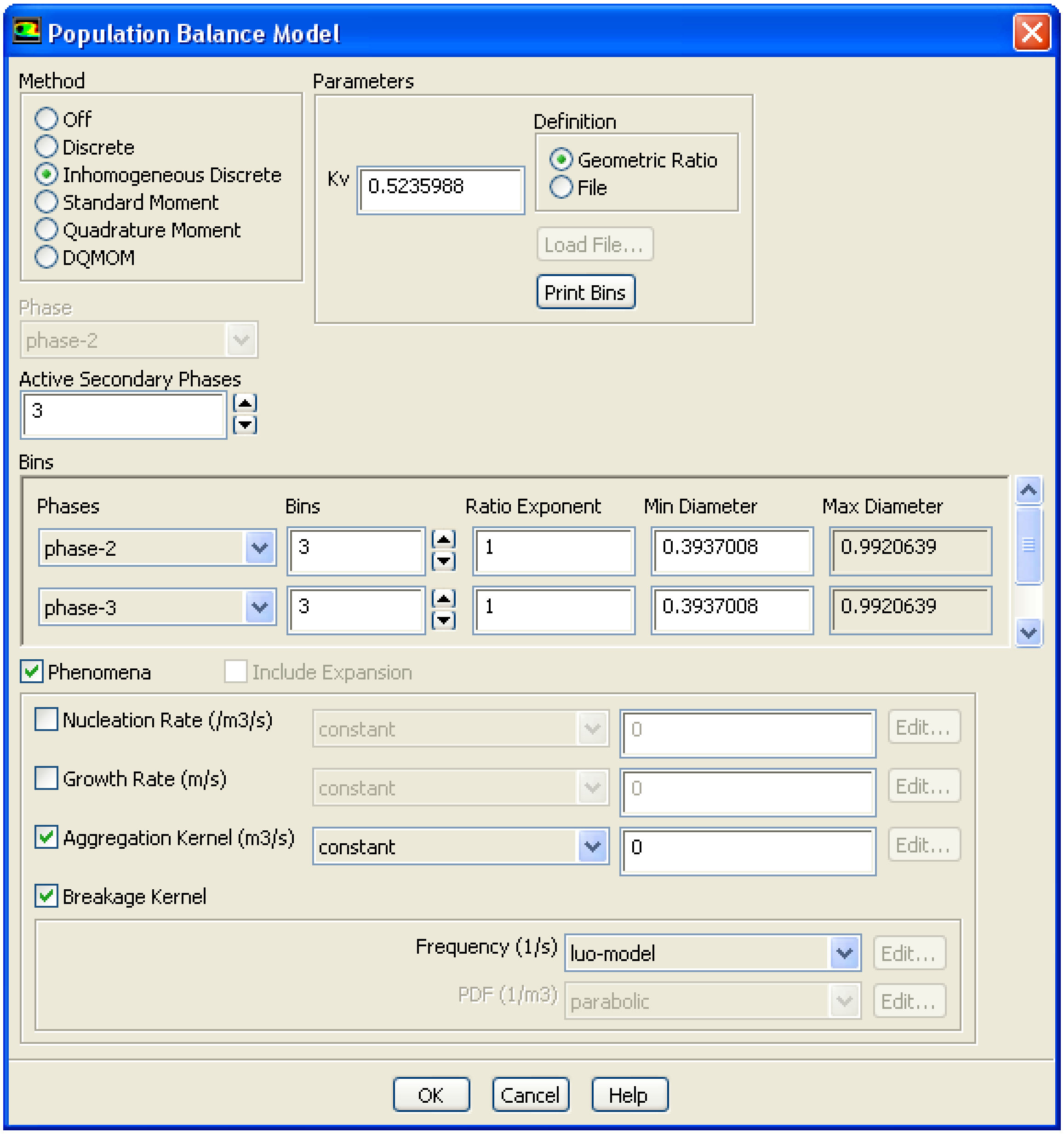
2. To enable the population balance model, follow the instructions in *Loading the Population Balance Module* (p. 29).

   Remember to enable the mixture or Eulerian multiphase model.

3. Open the **Population Balance Model** dialog box (*Figure 3.1* (p. 31)).

   Models → **Population Balance** → **Edit...**

Figure 3.1 The Population Balance Model Dialog Box



Population Balance Model

Method

- ○ Off
- ○ Discrete
- ● Inhomogeneous Discrete
- ○ Standard Moment
- ○ Quadrature Moment
- ○ DQMOM

Parameters

Kv 0.5235988

Definition

- ● Geometric Ratio
- ○ File

Phase phase-2

Active Secondary Phases 3

Bins

Phases	Bins	Ratio Exponent	Min Diameter	Max Diameter
phase-2	3	1	0.3937008	0.9920639
phase-3	3	1	0.3937008	0.9920639

☑ Phenomena □ Include Expansion

□ Nucleation Rate (/m3/s)	constant	0	Edit...
□ Growth Rate (m/s)	constant	0	Edit...
☑ Aggregation Kernel (m3/s)	constant	0	Edit...

☑ Breakage Kernel

Frequency (1/s)	luo-model	Edit...
PDF (1/m3)	parabolic	Edit...

4. Specify the population balance method under **Method**.

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

Kv

specifies the value for the particle volume coefficient K_V (as described in *Particle Growth and Dissolution* (p. 6)). By default, this coefficient has a value of $\pi/6$.

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 smallest to the largest diameter (one entry per line). Hence, you are not limited by the choices specified in the dialog box.

Bins
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 *Numerical Method* (p. 19)).

Min Diameter
specifies the minimum bin size $L_0 \equiv (V_0 / K_v)^{1/3}$.

Max Diameter
displays the maximum bin size, which is calculated internally.

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 **Inhomogeneous Discrete** under **Method**, you will specify the same parameters as for the **Discrete** model. Additionally, you can include more than one secondary phase in the bin definition. Enter the total number of **Active Secondary Phases** in your simulation.

> **Note**
>
> While reading bins through the **Load File...** option for the **Inhomogeneous Discrete** model, the corresponding phase name must be included, for example `(("air-1" (0.1 0.2 0.3))`

- 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**.
- If you selected **DQMOM** under **Method**, you will select the **DQMOM Phases** from the list. You will also specify the following **Parameters**:

Max Size
specifies the maximum size of the particle.

Min Size
specifies the minimum size of the particle.

Reference Length
is the reference particle size. Normally, the averaged size of the particle group should be sufficient.

Min VOF
is the minimum VOF, where the total volume fraction of the particle phases (participating in DQMOM computations) is below the minimum value; the source terms caused by the breakage and coalescence are not computed in that cell for the DQMOM and VOF equations.

Max VOF Change/Time Step
is the maximum VOF change in percentage for each DQMOM phase per time step, in order to smooth the convergence progress.

Generate DQMOM Values
enables you to generate DQMOM values from PDF, CDF, or Overall Moments files. Each of the file formats and the way to generate the values are discussed in *Generated DQMOM Values* (p. 37).

Note

The DQMOM method is restricted to a four-phase system, of which three secondary phases are directly involved in the DQMOM computation. Unsteady simulations are required to model breakage and coalescence and a well defined initial field is recommended, in which the abscissas are distinct. Only growth, breakage, and aggregation are the available phenomena. No nucleation is considered.

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

enables you to specify the nucleation rate ($\mathrm{particles} / \mathrm{m}^3 – \mathrm{s}$). 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.

Note

This option is not available when using the **Inhomogeneous Discrete** method.

Growth Rate

enables 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.

Note

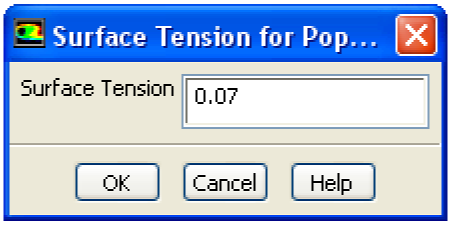
This option is not available when using the **Inhomogeneous Discrete** method.

Aggregation Kernel

enables 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 enable you to specify the surface tension (see *Figure 3.2* (p. 34)). The aggregation rate for the model will then be calculated based on Luo's aggregation kernel (as described in *Particle Birth and Death Due to Breakage and Aggregation* (p. 7)).

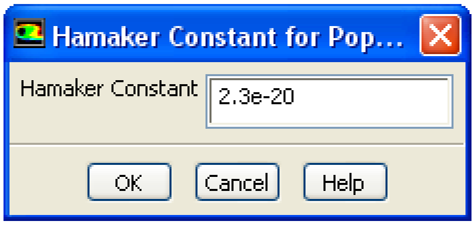
Figure 3.2 The Surface Tension for Population Balance Dialog Box



Surface Tension 0.07

- If you select **free-molecular-model**, then *Equation 2–38* (p. 16) is applied.
- If you select **turbulent-model**, the **Hamaker Constant for Population Balance** dialog box will open automatically to enable you to specify the Hamaker constant (see *Figure 3.3* (p. 34)). More information about this model is available in *Turbulent Aggregation Kernel* (p. 16).

Figure 3.3 The Hamaker Constant for Population Balance Dialog Box



Hamaker Constant 2.3e-20

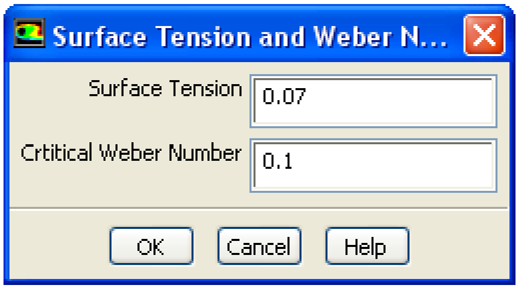
- If you have a user-defined function (UDF) that you want to use to model the aggregation rate, you can choose the **user-defined** option and specify the appropriate UDF.

Breakage Kernel

enables you to specify the particle breakage frequency ($\text{particles} / m^3\text{–}s$). You can select **constant**, **luo-model**, **lehr-model**, **ghadiri-model**, **laakkonen-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 enable you to specify the surface tension (see *Figure 3.2* (p. 34)). The frequency used in the breakage rate will then be calculated based on Luo’s breakage kernel (as described in *Particle Birth and Death Due to Breakage and Aggregation* (p. 7)).
- If you select **lehr-model**, the **Surface Tension and Weber Number** dialog box will open automatically to enable you to specify the surface tension and critical Weber number (see *Figure 3.4* (p. 34)). The frequency used in the breakage rate will then be calculated based on Lehr’s breakage kernel (as described in *Particle Birth and Death Due to Breakage and Aggregation* (p. 7)).

Figure 3.4 The Surface Tension and Weber Number Dialog Box

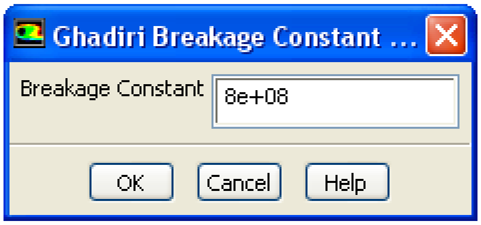


Surface Tension 0.07

Crtitical Weber Number 0.1

- If you select **ghadiri-model**, the **Ghadiri Breakage Constant for Population Balance** dialog box will open automatically to enable you to specify the breakage constant (see *Figure 3.5* (p. 35)). The frequency will then be calculated based on Ghadiri's breakage kernel (as described in *Particle Birth and Death Due to Breakage and Aggregation* (p. 7)).

Figure 3.5 The Ghadiri Breakage Constant for Population Balance Dialog Box



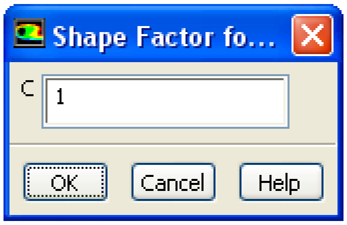
Breakage Constant 8e+08

- If you select **laakkonen-model**, the **Surface Tension for Population Balance** dialog box will open automatically to enable you to specify the **Surface Tension** and the constant **C2** (*Laakkonen Breakage Kernels* (p. 10)). The frequency will then be calculated based on Laakkonen's breakage kernel (as described in *Laakkonen Breakage Kernels* (p. 10)).
- If you have a user-defined function (UDF) that you want to use to model the frequency for the breakage rate, you can choose the **user-defined** option and specify the appropriate UDF.

If you selected **constant**, **ghadiri-model**, **laakkonen-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**, **laakkonen**, **generalized**, or **user-defined** :

- If you select **parabolic**, the **Shape Factor for Parabolic PDF** dialog box will open automatically to enable you to specify the shape factor **C** (see *Figure 3.6* (p. 35)). The PDF used in the breakage rate will then be calculated according to *Equation 2–20* (p. 11) (as described in *Particle Birth and Death Due to Breakage and Aggregation* (p. 7)).

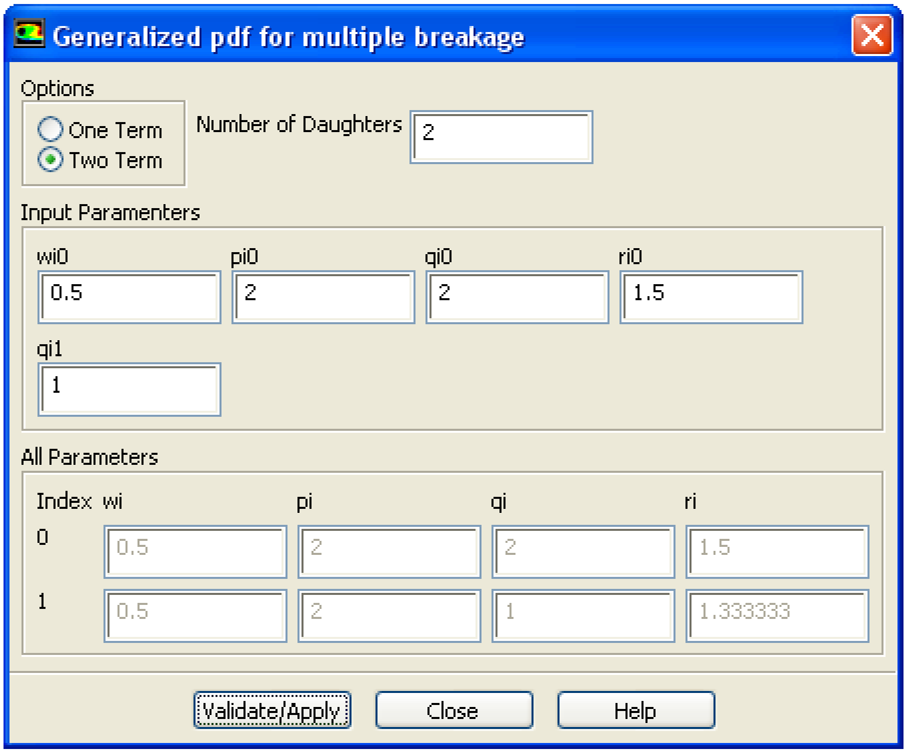
Figure 3.6 The Shape Factor for Parabolic PDF Dialog Box



C 1

- If you select **generalized**, the **Generalized pdf for multiple breakage** dialog box will open automatically (*Figure 3.7* (p. 36)).

Figure 3.7 The Generalized pdf for multiple breakage Dialog Box



Options

○ One Term
● Two Term

Number of Daughters: 2

Input Paramenters

wi0	pi0	qi0	ri0
0.5	2	2	1.5

qi1: 1

All Parameters

Index	wi	pi	qi	ri
0	0.5	2	2	1.5
1	0.5	2	1	1.333333

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 2–26* (p. 12) is 0 or 1, respectively.

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

c. Define the parameter(s) for *Equation 2–26* (p. 12) in the **Input Parameters** group box. When **One Term** is selected from the **Options** list, you must enter a value for **qi0**. When **Two Term** is selected from the **Options** list, you must enter values for **wi0**, **pi0**, **qi0**, **ri0**, and **qi1**. For information about appropriate values for these parameters to result in the daughter distributions shown in *Table 2.3: Daughter Distributions* (p. 12), see *Table 2.4: Daughter Distributions (cont.)* (p. 13).

> **Note**
>
> For equisized generalized pdf breakage distribution, the value for **qi0** should be set to 20.

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 *Equation 2–27* (p. 12) – *Equation 2–29* (p. 12).

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

UDFs for Population Balance Modeling (p. 53) 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 *Breakage Formulations for the Discrete Method* (p. 21).

7. Enable **Include Expansion** if you want to account for bubble expansion due to large changes in hydrostatic pressure.

   Note

   The secondary phase must be modeled as compressible. This option is currently available for Discrete and QMOM only.

8. Specify the boundary conditions for the solution variables.

   Boundary Conditions

   See *Defining Population Balance Boundary Conditions* (p. 41) below.

9. Specify the initial guess for the solution variables.

   Solution Initialization

10. Solve the problem and perform relevant postprocessing functions.

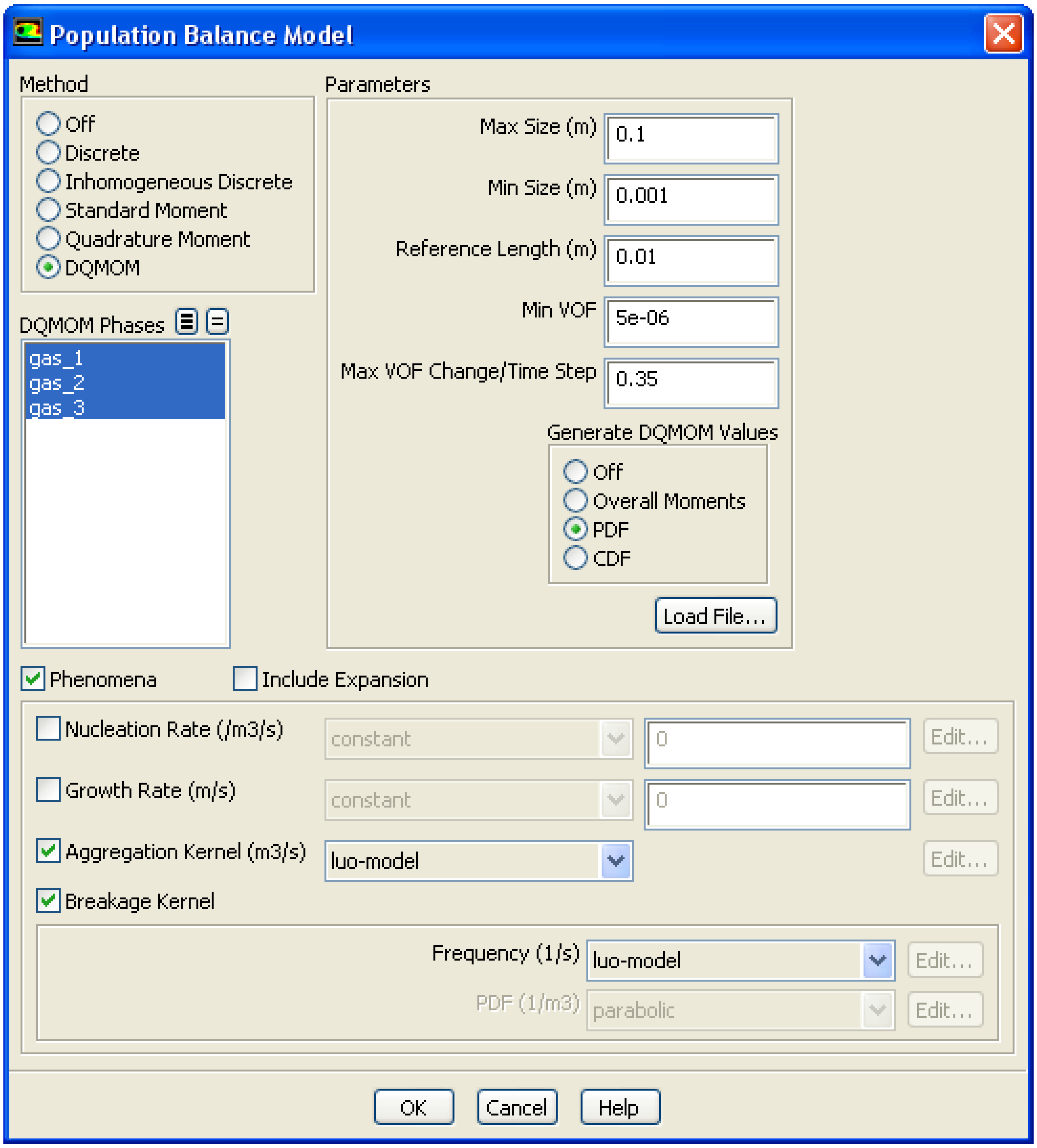
    Run Calculation

    See *Postprocessing for the Population Balance Model* (p. 49) for details about postprocessing.

3.3.1.1. Generated DQMOM Values

When using the **DQMOM** model, you have the option of generating DQMOM values from three different file formats. To do so, select **Overall Moments**, **PDF** or **CDF** under **Generate DQMOM Values** and then click the **Load File...** button. The **Select File** dialog box will open where you will select the appropriate file. Clicking **OK** in the **Select File** dialog box will result in the file being read and calculations carried out. The DQMOM values will be printed in the console.

Figure 3.8 The Population Balance Model Dialog Box for the DQMOM Model



Population Balance Model

Method
○ Off
○ Discrete
○ Inhomogeneous Discrete
○ Standard Moment
○ Quadrature Moment
● DQMOM

DQMOM Phases
gas_1
gas_2
gas_3

Parameters
Max Size (m) 0.1
Min Size (m) 0.001
Reference Length (m) 0.01
Min VOF 5e-06
Max VOF Change/Time Step 0.35

Generate DQMOM Values
○ Off
○ Overall Moments
● PDF
○ CDF

☑ Phenomena
□ Include Expansion

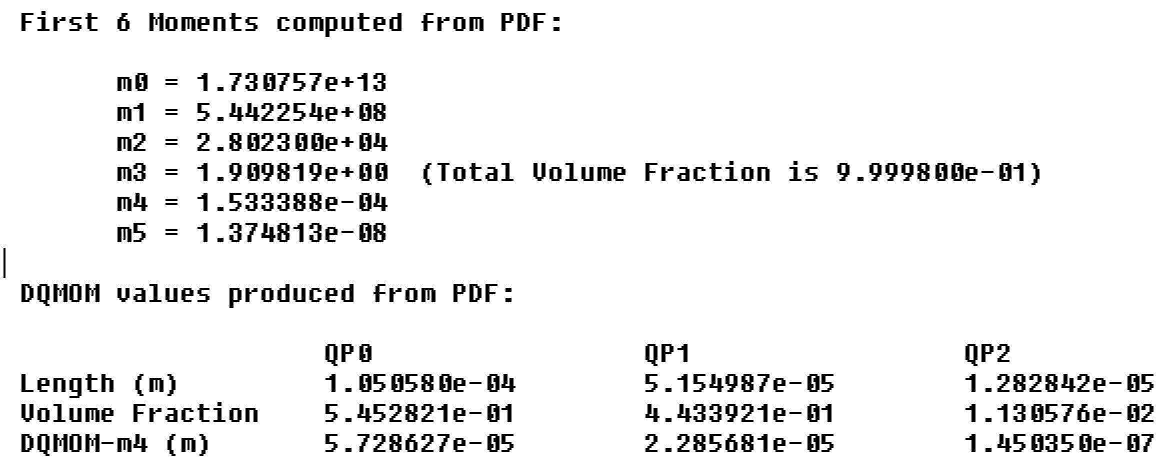
□ Nucleation Rate (/m3/s) constant 0
□ Growth Rate (m/s) constant 0
☑ Aggregation Kernel (m3/s) luo-model
☑ Breakage Kernel
Frequency (1/s) luo-model
PDF (1/m3) parabolic

DQMOM Values Produced From PDF, CDF Files, or Overall Moments for the Particles

- Three quadrature points are assumed, namely QP0, QP1, and QP2 (see *Figure 3.9* (p. 39)).
- `Length`, `Volume Fraction` , and `DQMOM-m4` values are given. The latter two can be used for initial fields of VOF and DQMOM as well as boundary conditions.
- For verification purposes, the first six moments are also given together with the total volume fraction of all particles from the PDF or CDF file. It is your responsibility to make sure that these values are correct, especially the total volume fraction.
- The definition of the first six moments of the particles is length based for the overall moments, described in *The Quadrature Method of Moments (QMOM)* (p. 24).
- In PDF or CDF format, the resultant volume fraction is normally given as unity. If you want the real particle volume fraction to be reflected in the mixture, the second column of the PDF or CDF data file need to be multiplied by the value of the real volume fraction. Another way is to multiply the

values of the generated DQMOM volume fraction and DQMOM-m4 using the real particle volume fraction.

Figure 3.9 DQMOM Values Produced From a PDF File



```
First 6 Moments computed from PDF:

      m0 = 1.730757e+13
      m1 = 5.442254e+08
      m2 = 2.802300e+04
      m3 = 1.909819e+00  (Total Volume Fraction is 9.999800e-01)
      m4 = 1.533388e-04
      m5 = 1.374813e-08

DQMOM values produced from PDF:

                   QP0                 QP1                 QP2
Length (m)         1.050580e-04        5.154987e-05        1.282842e-05
Volume Fraction    5.452821e-01        4.433921e-01        1.130576e-02
DQMOM-m4 (m)       5.728627e-05        2.285681e-05        1.450350e-07
```

PDF File Format

- The probability density function (PDF) is defined by the probability distribution of particles in terms of the volume fraction over the particle length (namely the diameter of the particle).
- The integration of PDF over all possible particle length (normally from 0 to the maximum diameter) shall give a value of unity or the real value of the volume fraction of all participating particles.
- The following file format is required (as shown below):
  - An integer number specified in the first line, indicating the number of data pairs to follow
  - The data in the 1st column specifying the length or diameter of particles in ascending order in meters (m)
  - The data in the 2nd column specifying the probability density function. Be aware that the integration of the PDF over the length shall result in a value of volume fraction for that particular particle length range

```
37
5e-6 0.000058e6
10e-6 0.000271e6
15e-6 0.000669e6
20e-6 0.001264e6
25e-6 0.002062e6
30e-6 0.003055e6
35e-6 0.004228e6
40e-6 0.005549e6
45e-6 0.006972e6
50e-6 0.008439e6
55e-6 0.00988e6
60e-6 0.011217e6
65e-6 0.012366e6
70e-6 0.013253e6
75e-6 0.013811e6
80e-6 0.013996e6
85e-6 0.013789e6
90e-6 0.013199e6
95e-6 0.012268e6
100e-6 0.011062e6
105e-6 0.009668e6
110e-6 0.00818e6
115e-6 0.006694e6
120e-6 0.00529e6
```

```
125e-6 0.004033e6
130e-6 0.002962e6
135e-6 0.002093e6
140e-6 0.00142e6
145e-6 0.000925e6
150e-6 0.000576e6
155e-6 0.000344e6
160e-6 0.000196e6
170e-6 0.000055e6
180e-6 0.000012e6
190e-6 0.000002e6
200e-6 0.
210e-6  0.
```

CDF File Format

- The cumulative density function (CDF) is defined as the integration of PDF over all possible particles up to the length L, resulting in a value of volume fraction for all particles less than length L.
- The value of the CDF at the maximum particle length/diameter shall be unity, or the real value of the volume fraction of all particles in the mixture.
- The following file format is required (as shown below):
  - An integer number specified in the first line, indicating the number of data pairs to follow
  - The data in the 1st column specifying the length or diameter of particles in ascending order in meters (m)
  - The data in the 2nd column specifying the cumulative density function in terms of the volume fraction of particles
  - $CDF \leq 1$ at the maximum particle length

```
37
5e-6 0
10e-6 0.109e-2
15e-6 0.16e-2
20e-6 0.175e-2
25e-6 0.208e-2
30e-6 0.304e-2
35e-6 0.614e-2
40e-6 1.5e-2
44e-6   3.e-2
45e-6   3.44e-2
50e-6 6.573e-2
55e-6 11.19e-2
60e-6 17.11e-2
65e-6 24.17e-2
70e-6 32.007e-2
75e-6 40.243e-2
80e-6 48.397e-2
85e-6 56.453e-2
90e-6 63.823e-2
95e-6 70.53e-2
100e-6 76.073e-2
105e-6 81e-2
110e-6 85.057e-2
115e-6 88.187e-2
120e-6 90.89e-2
125e-6 92.897e-2
130e-6 94.437e-2
135e-6 95.543e-2
140e-6 96.523e-2
145e-6 97.173e-2
150e-6 97.717e-2
155e-6 98.007e-2
160e-6 98.363e-2
170e-6 98.88e-2
180e-6 99.16e-2
```

```
190e-6 99.327e-2
200e-6 100.e-2
```

Overall Moments File Format

- The third option is to specify the first six moments for all particles as shown below.
- The following file format is required (as shown below):
  - An integer number specified in the first line, indicating the number of data (moments) to follow. By default, this shall be 6
  - six moments from moment-0 to moment-5 are given in that order. The definition of the first six moments is based on length as described in *The Quadrature Method of Moments (QMOM)* (p. 24)
  - As a check for moments, $Moment-3\,(m3) = {}^{1}\!/_{(K_v)\,(volume\_fraction)}$. In the values given below, volume fraction is assumed to be unity and K_vKv is assumed to be ${}^{\pi}\!/_{6}$

```
6
1.120556e+013
4.022475e+008
2.523370e+004
1.909857e+000
1.611191e-004
1.498663e-008
```

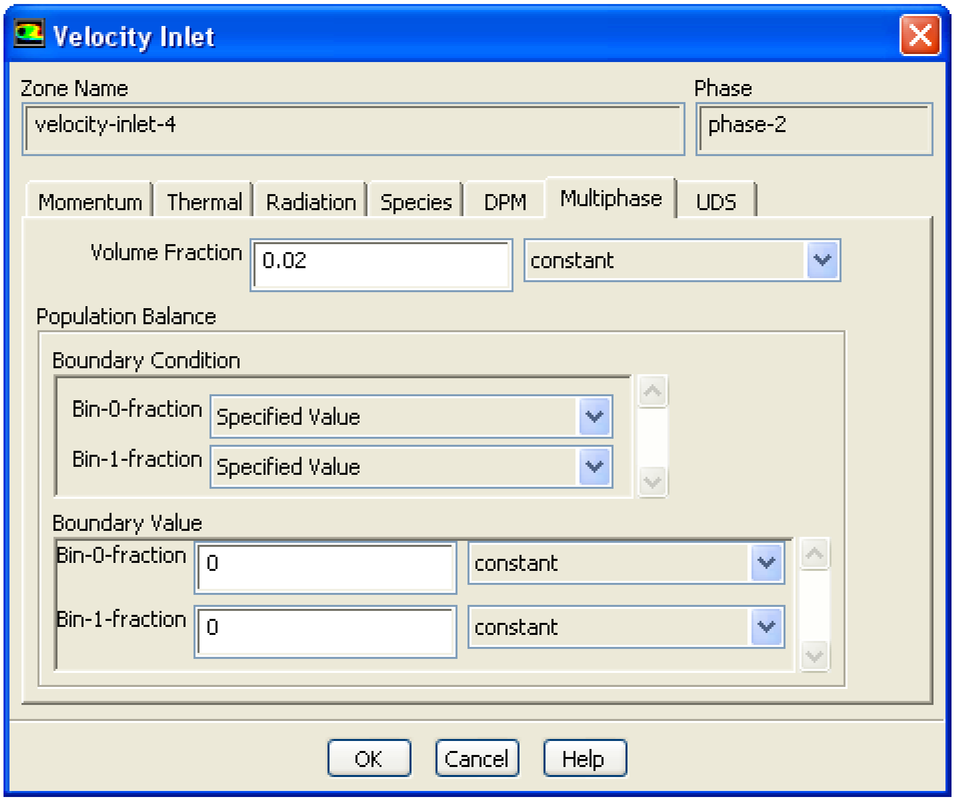
3.3.2. Defining Population Balance Boundary Conditions

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 (for example, *Figure 3.10* (p. 42)).

   Boundary Conditions

Figure 3.10 Specifying Inlet Boundary Conditions for the Population Balance Model



Velocity Inlet

Zone Name: velocity-inlet-4

Phase: phase-2

Volume Fraction: 0.02 constant

Population Balance

Boundary Condition

Bin-0-fraction: Specified Value

Bin-1-fraction: Specified Value

Boundary Value

Bin-0-fraction: 0 constant

Bin-1-fraction: 0 constant

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 (for example, **Bin-0**) are labeled according to the following:

   bin/moment **-** ith bin/moment

   where the ith *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 2–49* (p. 19) (for the discrete method) or m_k in *Equation 2–64* (p. 22) (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.

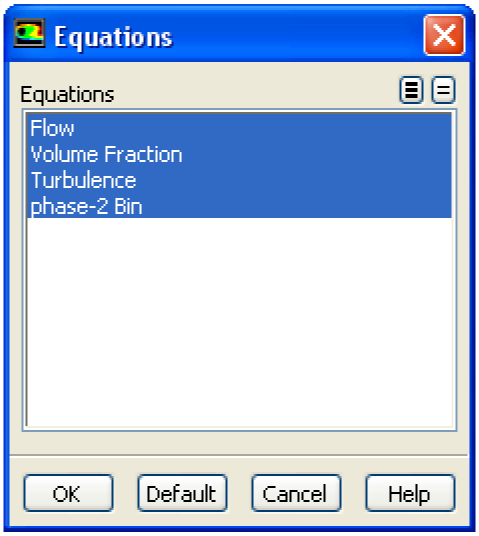
3.3.3. Specifying Population Balance Solution Controls

In the **Equations** dialog box (*Figure 3.11* (p. 43)), equations for each bin (for example, **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 3.11 The Equations Dialog Box



Equations

Equations

Flow
Volume Fraction
Turbulence
phase-2 Bin

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 2–71* (p. 23)) 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

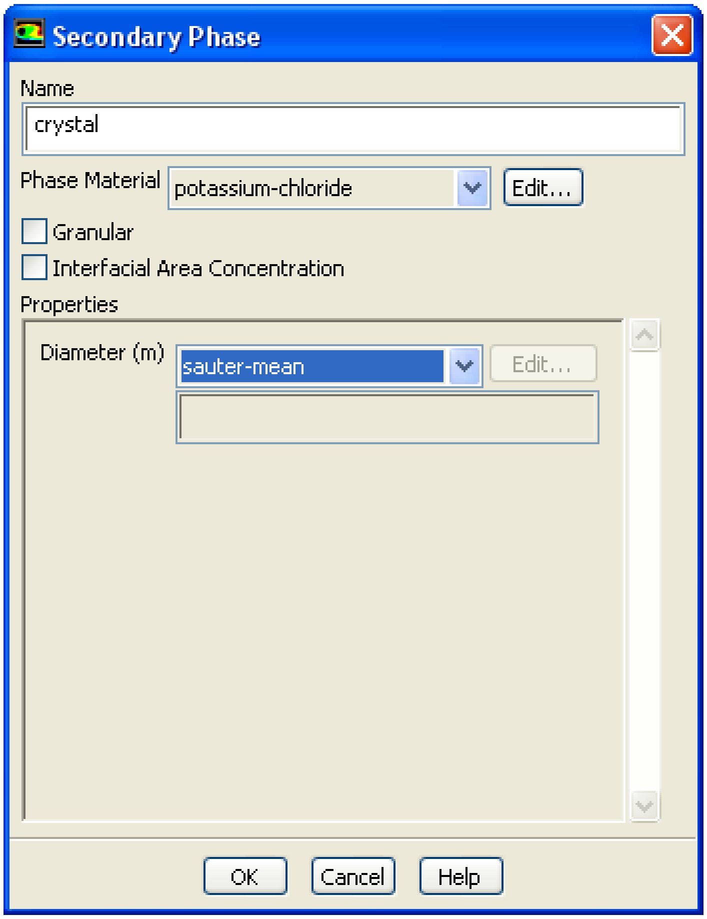
$$d_{32} = \frac{\sum N_i L_i^3}{\sum N_i L_i^2} \qquad \textbf{(3–1)}$$

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

Phases → **Secondary Phase** → **Edit...**

In the **Secondary Phase** dialog box (for example, *Figure 3.12* (p. 44)), 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 3.12 The Secondary Phase Dialog box for Hydrodynamic Coupling



Name: crystal

Phase Material: potassium-chloride

□ Granular

□ Interfacial Area Concentration

Diameter (m): sauter-mean

3.3.5. Specifying Interphase Mass Transfer Due to Nucleation and Growth

In applications that involve the creation, dissolution, or growth of particles (such as 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* (p. 61), 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 $\mathrm{kg} / \mathrm{m}^3\text{–}\mathrm{s}$) of particles of all sizes is then

$$\begin{aligned} \dot{m} &= 3\rho K_v \int_0^\infty L^2 G n(L)\, dL \\ &= \frac{1}{2}\rho K_a \int_0^\infty L^2 G n(L)\, dL \end{aligned} \qquad (3\text{–}2)$$

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

$$\begin{aligned}\dot{m} &= \rho\int_0^\infty G_v n\,(L)\,dL \\ &= \rho\int_0^\infty G_v n\,(V)\,dV \\ &= \rho\sum_i G_{v,i} N_i\end{aligned} \quad (3–3)$$

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

$$\dot{m} = \rho V_0 \dot{n}_0 + \sum_i \rho G_{v,i} N_i \quad (3–4)$$

Important

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 *UDFs for Population Balance Modeling* (p. 53) 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

$$\dot{m} = \frac{1}{2}\rho K_a G m_2 \quad (3–5)$$

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

$$\dot{m} = \frac{1}{2}\rho K_a \sum_i L_i^2 w_i G\,(L_i) \quad (3–6)$$

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

Important

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:

$$liquid\ (solvent) \rightarrow crystal\ (solute)$$

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 3.13* (p. 47)). 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. The nucleation and the growth rates calculated by the population balance kernels are used for mass transfer.

Click **OK** to save the settings.

Note

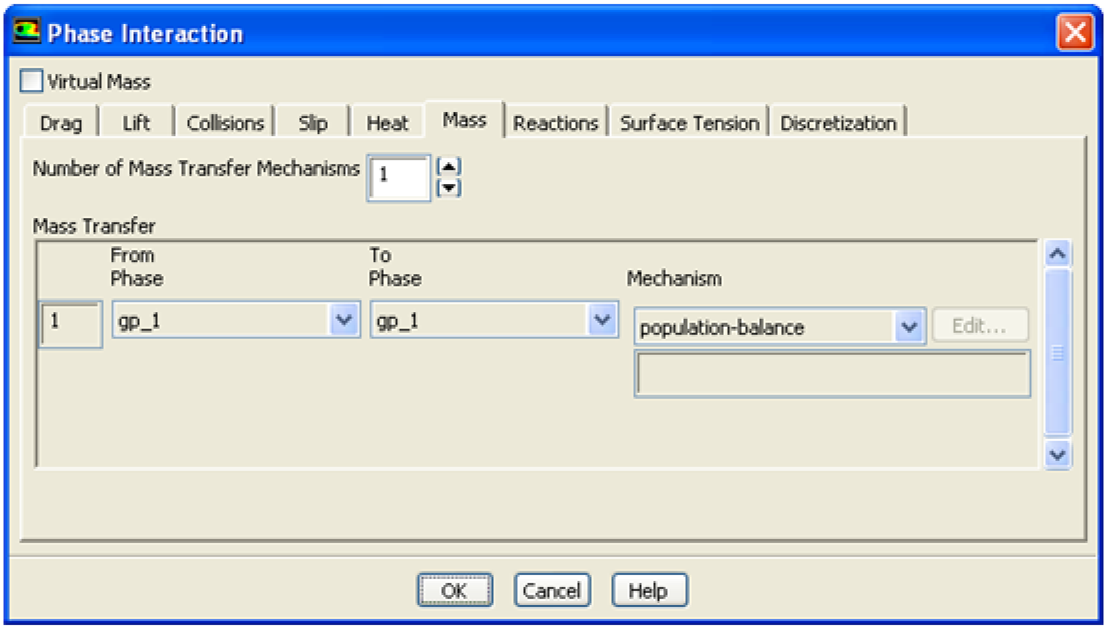
For the **Inhomogeneous Discrete** population balance model, where there is more than one secondary phase, you can select **population-balance** as the mechanism of mass transfer between the solvent phase (say for crystallization) and each of the solute phases defined under the **Inhomogeneous Discrete** population balance model.

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 3.14* (p. 47)). 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* (p. 61), will produce the same results.

For the **Inhomogeneous Discrete** population balance model involving nucleation and growth, you can select **population-balance** as the **Reaction Rate Function** for each heterogeneous reaction you have set up. To learn how to set up reactions, please go to Specifying Heterogeneous Reactions in the User's Guide.

Figure 3.13 The Phase Interaction Dialog Box for Non-reacting Species



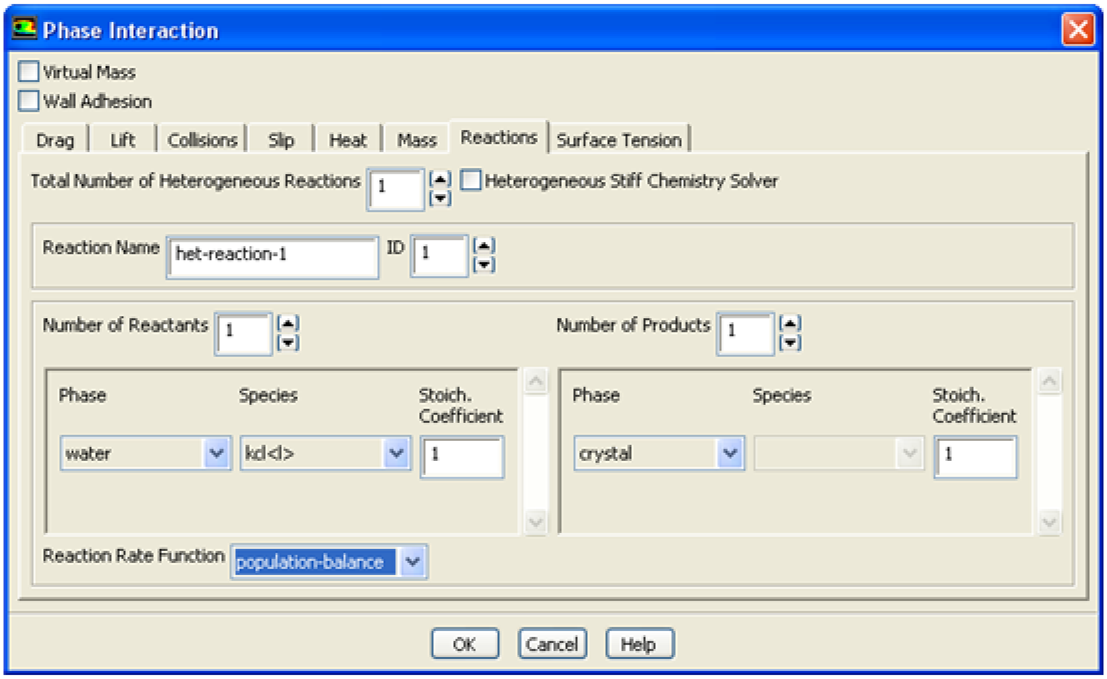
☐ Virtual Mass

Number of Mass Transfer Mechanisms 1

Mass Transfer

	From Phase	To Phase	Mechanism
1	gp_1	gp_1	population-balance

Figure 3.14 The Phase Interaction Dialog Box for a Heterogeneous Reaction



☐ Virtual Mass

☐ Wall Adhesion

Total Number of Heterogeneous Reactions 1 ☐ Heterogeneous Stiff Chemistry Solver

Reaction Name het-reaction-1 ID 1

Number of Reactants 1

Phase	Species	Stoich. Coefficient
water	kcl<l>	1

Number of Products 1

Phase	Species	Stoich. Coefficient
crystal		1

Reaction Rate Function population-balance

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.

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 ith size bin when using the discrete method. **Number density of Bin-i fraction** is the number density (N_i) in $\mathrm{particles} / \mathrm{m}^3$ for the ith size bin. **Moment-i** is the ith moment of the distribution when using the standard method of moments or the quadrature method of moments.

Important

Though the diffusion coefficients of the population variables (for example, **Diffusion Coef. of Bin-i fraction/Moment-i**) are available, they are set to zero because the diffusion term is not present in the population balance equations.

4.2. Reporting Derived Population Balance Variables

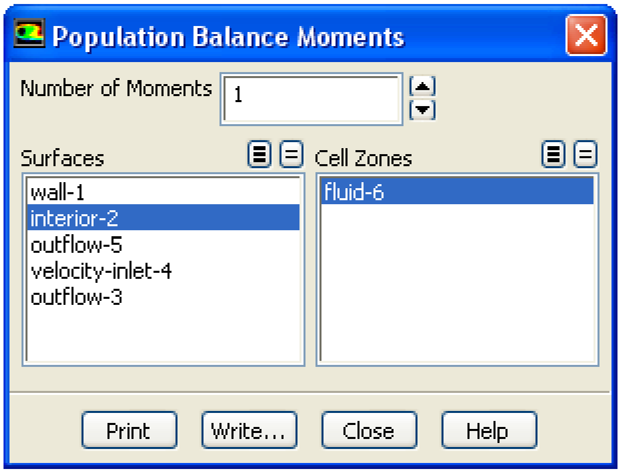
Two options are available in the **Report** menu that allow you to report computed moments and number density on selected surfaces or cell zones of the domain.

4.2.1. Computing Moments

You can compute moments for the population balance model using the **Population Balance Moments** dialog box (*Figure 4.1* (p. 50)).

Report → **Population Balance** → **Moments...**

Figure 4.1 The Population Balance Moments Dialog Box



Population Balance Moments

Number of Moments 1

Surfaces

- wall-1
- interior-2
- outflow-5
- velocity-inlet-4
- outflow-3

Cell Zones

- fluid-6

The steps for computing moments are as follows:

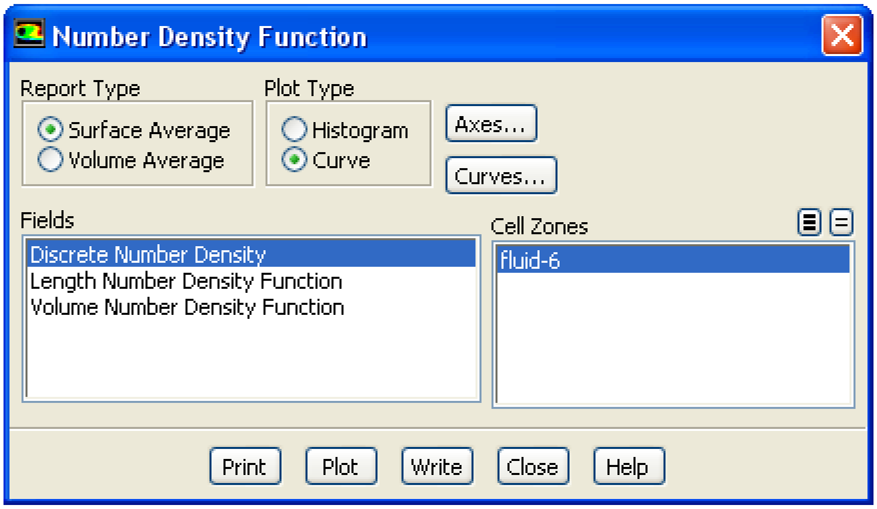
1. For the discrete method, specify the **Number of Moments**. For the SMM and QMOM, the number of moments is set equal to the number of moments that were solved, and therefore cannot be changed.
2. For a surface average, select the surface(s) on which to calculate the moments in the **Surfaces** list.
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.

4.2.2. 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* (p. 50)).

Report → Population Balance → Number Density...

Figure 4.2 The Number Density Function Dialog Box



Number Density Function

Report Type

● Surface Average
○ Volume Average

Plot Type

○ Histogram
● Curve

Fields

- Discrete Number Density
- Length Number Density Function
- Volume Number Density Function

Cell Zones

- fluid-6

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 ith 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

This chapter contains the following sections:

5.1. Population Balance Variables

The macros listed in *Table 5.1: Macros for Population Balance Variables Defined in* `sg_pb.h` (p. 53) 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 Macros for Population Balance Variables Defined in `sg_pb.h`

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

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.

5.2.1. DEFINE_PB_BREAK_UP_RATE_FREQ

You can use the DEFINE_PB_BREAK_UP_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.

5.2.1.1. Usage

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

Argument Type	Description
char name	UDF name
cell_t cell	Cell index
Thread *thread	Pointer to the secondary phase thread associated with d_1
real d_1	Parent particle diameter or length

Function returns

real

There are four arguments to DEFINE_PB_BREAK_UP_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.

5.2.1.2. Example

Included below is an example of a UDF that defines a breakage frequency (see *Particle Birth and Death Due to Breakage and Aggregation* (p. 7)) that is based on the work of Tavlarides [4] (p. 65), such that

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

where C_1 and C_2 are constants, ε is the dissipation rate, d is the parent diameter, σ is the surface tension, α is the volume fraction of the dispersed phase, and ρ_l is the density of the primary phase.

```
/*************************************************************************
 UDF that computes the particle breakage frequency
 *************************************************************************/

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

 DEFINE_PB_BREAK_UP_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.+alpha)*(1.+alpha))/(rho_d*pow(epsi,2./3.)*pow(d_1, 5./3.));
     return C1*f1*exp(f2);
 }
```

5.2.2. DEFINE_PB_BREAK_UP_RATE_PDF

You can use the DEFINE_PB_BREAK_UP_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.

5.2.2.1. Usage

```
DEFINE_PB_BREAK_UP_RATE_PDF(name, cell, thread, d_1, thread_2, d_2)
```

Argument Type	Description
char name	UDF name
cell_t cell	Cell index
Thread *thread	Pointer to the secondary phase thread associated with d_1
real d_1	Parent particle diameter or length
Thread *thread_2	Pointer to the secondary phase thread associated with d_2
real d_2	Diameter of one of the daughter particles after breakage; the second daughter particle diameter is calculated by conservation of particle volume

Function returns

real

There are six arguments to DEFINE_PB_BREAK_UP_RATE_PDF: name, cell, thread, d_1, thread_2, and d_2. You will supply name, the name of the UDF. cell, thread, d_1, thread_2, and d_2 are variables that are passed by the ANSYS FLUENT solver to your UDF.

Note

thread and thread_2 are the same for the Discrete, QMOM and SMM models. They may be the same or different depending on whether d_1 and d_2 belong to the same phase or different phases for the Inhomogeneous model.

5.2.2.2. Example

Included below is an example of a UDF that defines a breakage PDF (see *Particle Birth and Death Due to Breakage and Aggregation* (p. 7)) that is parabolic, as defined in *Equation 2–20* (p. 11).

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

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

 DEFINE_PB_BREAK_UP_RATE_PDF(break_up_pdf_par, cell, thread, d_1, thread_2, 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.

5.2.3.1. Usage

```
DEFINE_PB_COALESCENCE_RATE(name, cell, thread, d_1, thread_2, d_2)
```

Argument Type	Description
char name	UDF name
cell_t cell	Cell index
Thread *thread	Pointer to the secondary phase thread associated with d_1
Thread *thread_2	Pointer to the secondary phase thread associated with d_2
real d_1, d_2	Diameters of the two colliding particles

Function returns

real

There are six arguments to DEFINE_PB_COALESCENCE_RATE: name, cell, thread, d_1, thread_2, 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.

Note

thread and thread_2 are the same for the Discrete, QMOM and SMM models. They may be the same or different depending on whether d_1 and d_2 belong to the same phase or different phases for the Inhomogeneous model.

5.2.3.2. 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.

```
/*************************************************************************
 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,thread_2,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.

5.2.4.1. 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.

5.2.4.2. 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
 *************************************************************************/

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

Important

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.

5.2.5.1. Usage

```
DEFINE_PB_GROWTH_RATE(name, cell, thread,d_i)
```

Argument Type	Description
char name	UDF name
cell_t cell	Cell index
Thread *thread	Pointer to the secondary phase thread
real d_i	Particle diameter or length

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.

5.2.5.2. 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 $K_g = 2.8 \times 10^{-8}$ m/s and $N_g = 1$.

```
/*************************************************************************
 UDF that computes the particle growth rate
 *************************************************************************/

 #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;
 }
```

Important

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_BREAK_UP_RATE_FREQ`, `DEFINE_PB_BREAK_UP_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 (for example, `agg_kernel`) will become visible and selectable in the appropriate drop-down list under **Phenomena** in the **Population Balance Model** dialog box (*Figure 3.1* (p. 31)).

Appendix A. DEFINE_HET_RXN_RATE Macro

This appendix discusses the DEFINE_HET_RXN_RATE macro:

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 (that is, the 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)
```

Argument Type	Description
char name	UDF name.
cell_t c	Cell index.
Thread *t	Cell thread (mixture level) on which heterogeneous reaction rate is to be applied.
Hetero_Reaction *r	Pointer to data structure that represents the current heterogeneous reaction (see sg_mphase.h).
real mw[MAX_PHASES][MAX_SPE_EQNS]	Matrix of species molecular weights. mw[i][j] will give molecular weight of species with ID j in phase with index i. For phase that has fluid material, the molecular weight can be accessed as mw[i][0].
real yi[MAX_PHASES][MAX_SPE_EQNS]	Matrix of species mass fractions. yi[i][j] will give molecular weight of species with ID j in phase with index i. For phase that has fluid material, yi[i][0] will be 1.

`real *rr`	Pointer to laminar reaction rate.
`real *rr_t`	Currently not used. Provided for future use.

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.

```
#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);
          *rr += stoich_reactant[i]*log(ci);
       }

       *rr += log(Arrhenius + SMALL_S) -
                   E_Activation/(UNIVERSAL_GAS_CONSTANT*T);

    /* 1.e-40 rr 1.e40 */
    *rr = MAX(*rr,-40);
    *rr = MIN(*rr,40);

    *rr = exp(*rr);
 }
```

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 ANSYS FLUENT UDF Manual for details), the name that you specified in the `DEFINE` macro argument (for example, `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**.)

Bibliography

[1] J. Abrahamson. *Collision Rates of Small Particles in a Vigorously Turbulent Fluid*. *Chemical Engineering Science*. 30. 1371–1379. 1975.

[2] L. Austin, K. Shoji, V. Bhatia, V. Jindal, K. Savage, and R. Klimpel*Some Results on the Description of Size Distribution as a Rate Process in Various Mills*. *Industrial Engineering and Chemical Process Design Devices*. 15(1). 187–196. 1976.

[3] J. Baldyga and W. Orciuch. *Barium Sulfate Precipitation in a Pipe – An Experimental Study and CFD Modeling*. *Chemical Engineering Science*. 56. 2435–2444. 2001.

[4] C. A. Coulaloglou and L. L. Tavlarides. *Description of Interaction Processes in Agitated Liquid-Liquid Dispersions*. *Chem. Eng. Sci*. 32. 1289–1297. 1977.

[5] H. Dette and W. J. Studden. *The Theory of Canonical Moments with Applications in Statistics, Probability, and Analysis*. John Wiley & Sons. New York, NY1997.

[6] R. B. Diemer and J. H. Olson. *A Moment Methodology for Coagulation and Breakage Problems Part 3 — Generalized Daughter Distribution Functions*. *Chemical Engineering Science*. 57(19). 4187–4198. 2002.

[7] M. Ghadiri and Z. Zhang. *Impact Attrition of Particulate Solids Part 1. A Theoretical Model of Chipping*. *Chemical Engineering Science*. 57. 3659–3669. 2002.

[8] R. G. Gordon. *Error Bounds in Equilibrium Statistical Mechanics*. *Journal of Mathematical Physics*. 56. 655–633. 1968.

[9] K. Higashitani, K. Yamauchi, Y. Matsuno, and G. Hosokawa. *Turbulent Coagulation of Particles Dispersed in a Viscous Fluid*. *Chemical Engineering Journal Japan*. 16(4). 299–304. 1983.

[10] M. J. Hounslow, R. L. Ryall, and V. R. Marshall. *A Discretized Population Balance for Nucleation, Growth, and Aggregation*. *AIChE Journal*. 34(11). 1821–1832. 1988.

[11] M. A. Hsia and L. L. Tavlarides. *Simulation analysis of drop breakage, coalescence and micro-mixing in liquid–liquid stirred tanks*. *Chemical Engineering Journal*. 26(3). 189–199. 1983.

[12] M. Kostoglou, S. Dovas, and A. J. Karabelas. *On the Steady-State Size Distribution of Dispersions in Breakage Processes*. *Chemical Engineering Science*. 52(8). 1285–1299. 1997.

[13] S. Kumar and D. Ramkrishna. *On the Solution of Population Balance Equations by Discretization - I., A Fixed Pivot Technique*. *Chemical Engineering Science*. 51(8). 1311–1332. 1996.

[14] Hagesaether L., Jakobsen H.A.1, and Svendsen H.F. *A Model for Turbulent Binary Breakup of Dispersed Fluid Particles*. *Chemical Engineering Science*. 57(16). 3251–3267. 2002.

[15] F. Lehr, M. Millies, and D. Mewes. *Bubble-Size Distributions and Flow Fields in Bubble Columns*. *AIChE Journal*. 8(11). 2426–2443. 2002.

[16] J. D.Litster, D. J.Smit, and M. J.Hounslow. *Adjustable Discretization Population Balance for Growth and Aggregation*. *AIChE Journal*. 41(3). 591–603. 1995.

[17] H. Luo. *Coalescence, Breakup and Liquid Circulation in Bubble Column Reactors*. *PhD thesis from the Norwegian Institute of Technology*. Trondheim,Norway1993.

[18] H. Luo and H. F. Svendsen.*Theoretical Model for Drop and Bubble Breakup in Turbulent Dispersions*. *AIChE Journal*. 42(5). 1225–1233,. 1996.

[19] D. L. Marchisio, R. D. Virgil, and R. O. Fox.*Quadrature Method of Moments for Aggregation-Breakage Processes*. *Journal of Colloid and Interface Science*. 258. 322–334. 2003.

[20] B. J. McCoy and M. Wang.*Continuous-mixture Fragmentation Kinetics Particle Size Reduction and Molecular Cracking*. *Chemical Engineering Science*. 49(22). 3773–3785. 1994.

[21] R. McGraw.*Description of Aerosol Dynamics by the Quadrature Method of Moments*. *Aerosol Science and Technology*. 27. 255–265. 1997.

[22] R. Moreno-Atanasio and M. Ghadiri.*Mechanistic Analysis and Computer Simulation of Impact Breakage of Agglomerates Effect of Surface Energy*. *Chemical Engineering Science*. 61(8). 2476–2481. 2006.

[23] S. B. Pope.*Probability Distributions of Scalars in Turbulent Shear Flow*. *Turbulent Shear Flows*. volume 2. pages 7–16. 1979.

[24] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery.*Numerical Recipes*. Cambridge University Press. Cambridge, England,1992.

[25] D. Ramkrishna.*Population Balances Theory and Applications to Particulate Systems in Engineering*. Academic Press. San Diego, CA,2000.

[26] A. D. Randolph and M. A. Larson.*Theory of Particulate Processes. Analysis and Techniques of Continuous Crystallization*. Academic Press. San Diego, CA1971.

[27] P.G. Saffman and J.S. Turner. *On the Collision of Droplets in Turbulent Clouds*. *Journal of Fluid Mechanics,*. 1. 16–30. 1956.

[28] M. Schmoluchowski.*Versuch Einen Mathematischen Theorie der Koagulationstechnik Kolloider Lösungen*. *Zeitschrift für Physikalische Chemie*. 92. 129–168. 1917.

[29] R. D.Vigil and R. M.Ziff.*On the Stability of Coagulation–Fragmentation Population Balances*. *Journal of Colloids and Interface Science*. 133(1). 257–264. 1989.

[30] R. Fan, D.L. Marchisio and R.O. Fox. *Application of the Direct Quadrature Method of Moments to Polydisperse Gas-solid Fluidized Beds*. *Power Technology*. 139. 7–20. 2004.

Index