

# Granular Flow Module

User's Guide

# Granular Flow Module User's Guide

© 1998–2025 COMSOL

Protected by patents listed on [www.comsol.com/patents](http://www.comsol.com/patents), or see Help > About COMSOL Multiphysics on the File menu in the COMSOL Desktop for less detailed lists of U.S. Patents that may apply. Patents pending.

This Documentation and the Programs described herein are furnished under the COMSOL Software License Agreement ([www.comsol.com/sla](http://www.comsol.com/sla)) and may be used or copied only under the terms of the license agreement.

Support for implementation of the ODB++ Format was provided by Mentor Graphics Corporation pursuant to the ODB++ Solutions Development Partnership General Terms and Conditions. ODB++ is a trademark of Mentor Graphics Corporation.

COMSOL, the COMSOL logo, COMSOL Multiphysics, COMSOL Desktop, COMSOL Compiler, COMSOL Server, and LiveLink are either registered trademarks or trademarks of COMSOL AB. All other trademarks are the property of their respective owners, and COMSOL AB and its subsidiaries and products are not affiliated with, endorsed by, sponsored by, or supported by those trademark owners. For a list of such trademark owners, see [www.comsol.com/trademarks](http://www.comsol.com/trademarks).

Version: COMSOL 6.4

## Contact Information

Visit the Contact COMSOL page at [www.comsol.com/contact](http://www.comsol.com/contact) to submit general inquiries or search for an address and phone number. You can also visit the Worldwide Sales Offices page at [www.comsol.com/contact/offices](http://www.comsol.com/contact/offices) for address and contact information.

If you need to contact Support, an online request form is located on the COMSOL Access page at [www.comsol.com/support/case](http://www.comsol.com/support/case). Useful links:

- Support Center: [www.comsol.com/support](http://www.comsol.com/support)
- Product Download: [www.comsol.com/product-download](http://www.comsol.com/product-download)
- Product Updates: [www.comsol.com/product-update](http://www.comsol.com/product-update)
- COMSOL Blog: [www.comsol.com/blogs](http://www.comsol.com/blogs)
- Discussion Forum: [www.comsol.com/forum](http://www.comsol.com/forum)
- Events: [www.comsol.com/events](http://www.comsol.com/events)
- COMSOL Video Gallery: [www.comsol.com/videos](http://www.comsol.com/videos)
- Support Knowledge Base: [www.comsol.com/support/knowledgebase](http://www.comsol.com/support/knowledgebase)
- Learning Center: [www.comsol.com/support/learning-center](http://www.comsol.com/support/learning-center)

Part number: CM025401

# C o n t e n t s

## Chapter 1: Introduction

<b>About the Granular Flow Module</b>	<b>6</b>
The Granular Flow Module Physics Interface Guide . . . . .	6
Common Physics Interface and Feature Settings and Nodes . . . . .	6
Where Do I Access the Documentation and Application Libraries? . . . . .	7
<b>Overview of the User's Guide</b>	<b>11</b>

## Chapter 2: Modeling Granular Flow

<b>Introduction to Granular Flow Modeling</b>	<b>14</b>
<b>General Tools for Granular Flow Modeling</b>	<b>15</b>
Special Variables . . . . .	15
Nonlocal Couplings . . . . .	16
Sampling from Random Number Distributions . . . . .	18
Study Setup . . . . .	20
The Grain Dataset . . . . .	20

## Chapter 3: The Granular Flow Interface

<b>The Granular Flow Interface</b>	<b>22</b>
List of Granular Flow Interface Physics Features . . . . .	26
Grain Properties . . . . .	27
Wall . . . . .	29
Contact Between Grains . . . . .	31
Contact with Walls . . . . .	34
Gravity . . . . .	38
Release . . . . .	39
Inlet . . . . .	43

Release from Grid . . . . .	43
Release from Data File. . . . .	46
Outlet . . . . .	48
Force . . . . .	48
Torque . . . . .	48
Auxiliary Dependent Variable . . . . .	49
Periodic Condition . . . . .	50
Grain Counter. . . . .	51
Bounding Box . . . . .	51
Heat Source . . . . .	52
Convective Heat Transfer . . . . .	52
Accumulator (Boundary) . . . . .	53
Force Accumulator . . . . .	55
Accumulator (Domain) . . . . .	55
<b>Theory for the Granular Flow Interface</b>	<b>57</b>
Kinematics of Grains . . . . .	57
Contact Force: Linear Elastic Model. . . . .	59
Contact Force: Hertz–MD (Mindlin and Deresiewicz) Model . . . . .	62
Contact Force: Hertz–MD with Adhesion Model. . . . .	64
Contact Force: van der Waals Force . . . . .	65
Contact Force: Coulomb’s Criterion . . . . .	66
Rotational Resistance Theory . . . . .	67
Computing Grain Temperature . . . . .	70
Contact Search Theory . . . . .	73
Initial Conditions . . . . .	75
Boundary Conditions . . . . .	75
Auxiliary Dependent Variables Theory. . . . .	77
Accumulator Theory: Domains . . . . .	77
Accumulator Theory: Boundaries . . . . .	78
Force Accumulator Theory . . . . .	80
Time Step Size. . . . .	81
References . . . . .	82



# Introduction

This user's guide describes the Granular Flow Module, an optional add-on package for COMSOL Multiphysics® designed to compute grain trajectories. The grains can interact with each other and with boundaries, and their motion can be affected by external fields, which can be user defined or solved for by other physics interfaces.

The *Granular Flow Module User's Guide* introduces the modeling stages in COMSOL Multiphysics® and this module and serves as a reference for more advanced modeling techniques and details about the physics interfaces.

In this chapter:

- [About the Granular Flow Module](#)
- [Overview of the User's Guide](#)

# About the Granular Flow Module

In this section:

- [The Granular Flow Module Physics Interface Guide](#)
- [Common Physics Interface and Feature Settings and Nodes](#)
- [Where Do I Access the Documentation and Application Libraries?](#)

## *The Granular Flow Module Physics Interface Guide*

---

The Granular Flow Module extends the functionality of the physics interfaces of the base package for COMSOL Multiphysics. The details of the physics interfaces and study types for the Granular Flow Module are listed in the table. The functionality of the COMSOL Multiphysics base package is given in the *COMSOL Multiphysics Reference Manual*.

---

In the *COMSOL Multiphysics Reference Manual*:



- [Studies and Solvers](#)
  - [The Physics Interfaces](#)
  - For a list of all the core physics interfaces included with a COMSOL Multiphysics license, see [Physics Interface Guide](#).
- 

## *Common Physics Interface and Feature Settings and Nodes*

---

There are several common settings and sections available for the physics interfaces and feature nodes. Some of these sections also have similar settings or are implemented in the same way no matter the physics interface or feature being used. There are also some physics feature nodes that display in COMSOL Multiphysics.

In each module's documentation, only unique or extra information is included; standard information and procedures are centralized in the *COMSOL Multiphysics Reference Manual*.



---

In the *COMSOL Multiphysics Reference Manual* see [Table 2-4](#) for links to common sections and [Table 2-5](#) to common feature nodes. You can also search for information: press F1 to open the **Help** window or Ctrl+F1 to open the **Documentation** window.

---

### *Where Do I Access the Documentation and Application Libraries?*

---

A number of online resources have more information about COMSOL, including licensing and technical information. The electronic documentation, topic-based (or context-based) help, and the Application Libraries are all accessed through the COMSOL Desktop.



---

If you are reading the documentation as a PDF file on your computer, the [blue links](#) do not work to open an application or content referenced in a different guide. However, if you are using the Help system in COMSOL Multiphysics, these links work to open other modules, application examples, and documentation sets.


---

## **THE DOCUMENTATION AND ONLINE HELP**




The *COMSOL Multiphysics Reference Manual* describes the core physics interfaces and functionality included with the COMSOL Multiphysics license. This book also has instructions on how to use COMSOL Multiphysics and how to access the electronic Documentation and Help content.

### *Opening Topic-Based Help*

The Help window is useful as it is connected to the features in the COMSOL Desktop. To learn more about a node in the Model Builder, or a window on the Desktop, click to highlight a node or window, then press F1 to open the Help window, which then




displays information about that feature (or click a node in the Model Builder followed by the **Help** button (  ). This is called *topic-based* (or *context*) *help*.

---

	<p>To open the <b>Help</b> window:</p> <ul style="list-style-type: none"><li>• In the <b>Model Builder</b>, <b>Application Builder</b>, or <b>Physics Builder</b>, click a node or window and then press F1.</li><li>• On any toolbar (for example, <b>Home</b>, <b>Definitions</b>, or <b>Geometry</b>), hover the mouse over a button (for example, <b>Add Physics</b> or <b>Build All</b>) and then press F1.</li><li>• From the <b>File</b> menu, click <b>Help</b> (  ).</li><li>• In the upper-right corner of the COMSOL Desktop, click the <b>Help</b> (  ) button.</li></ul>
---	---

---



---

	<p>To open the <b>Help</b> window:</p> <ul style="list-style-type: none"><li>• In the <b>Model Builder</b> or <b>Physics Builder</b>, click a node or window and then press F1.</li><li>• In the main toolbar, click the <b>Help</b> (  ) button.</li><li>• From the main menu, select <b>Help &gt; Help</b>.</li></ul>
	

---




### *Opening the Documentation Window*

---

	<p>To open the <b>Documentation</b> window:</p> <ul style="list-style-type: none"><li>• Press Ctrl+F1.</li><li>• From the <b>File</b> menu, select <b>Help &gt; Documentation</b> (  ).</li></ul>
---	--

---

---

	<p>To open the <b>Documentation</b> window:</p> <ul style="list-style-type: none"><li>• Press Ctrl+F1.</li><li>• In the main toolbar, click the <b>Documentation</b> (  ) button.</li><li>• From the main menu, select <b>Help &gt; Documentation</b>.</li></ul>
	

---

## THE APPLICATION LIBRARIES WINDOW

Each model or application includes documentation with the theoretical background and step-by-step instructions to create a model or application. The models and applications are available in COMSOL Multiphysics as MPH files that you can open for further investigation. You can use the step-by-step instructions and the actual models as templates for your own modeling. In most models, SI units are used to describe the relevant properties, parameters, and dimensions, but other unit systems are available.

Once the Application Libraries window is opened, you can search by name or browse under a module folder name. Click to view a summary of the model or application and its properties, including options to open it or its associated PDF document.



[The Application Libraries Window](#) in the *COMSOL Multiphysics Reference Manual*.

---


### Opening the Application Libraries Window

To open the **Application Libraries** window (  ):

---

Win

From the **File** menu, select **Application Libraries**.


To include the latest versions of model examples, from the **File > Help** menu select (  ) **Update COMSOL Application Libraries**.

---

Mac

From the **File** or **Windows** menu, select **Application Libraries**.

Linux

To include the latest versions of model examples, from the **Help** menu select (  ) **Update COMSOL Application Libraries**.

---

## CONTACTING COMSOL BY EMAIL

For general product information, contact COMSOL at [info@comsol.com](mailto:info@comsol.com).

## COMSOL ACCESS AND TECHNICAL SUPPORT

To receive technical support from COMSOL for the COMSOL products, please contact your local COMSOL representative or send your questions to [support@comsol.com](mailto:support@comsol.com). An automatic notification and a case number will be sent to you

by email. You can also access technical support, software updates, license information, and other resources by registering for a COMSOL Access account.

#### **COMSOL ONLINE RESOURCES**

COMSOL website	<a href="http://www.comsol.com">www.comsol.com</a>
Contact COMSOL	<a href="http://www.comsol.com/contact">www.comsol.com/contact</a>
COMSOL Access	<a href="http://www.comsol.com/access">www.comsol.com/access</a>
Support Center	<a href="http://www.comsol.com/support">www.comsol.com/support</a>
Product Download	<a href="http://www.comsol.com/product-download">www.comsol.com/product-download</a>
Product Updates	<a href="http://www.comsol.com/product-update">www.comsol.com/product-update</a>
COMSOL Blog	<a href="http://www.comsol.com/blogs">www.comsol.com/blogs</a>
Discussion Forum	<a href="http://www.comsol.com/forum">www.comsol.com/forum</a>
Events	<a href="http://www.comsol.com/events">www.comsol.com/events</a>
COMSOL Application Gallery	<a href="http://www.comsol.com/models">www.comsol.com/models</a>
COMSOL Video Gallery	<a href="http://www.comsol.com/videos">www.comsol.com/videos</a>
Learning Center	<a href="http://www.comsol.com/support/learning-center">www.comsol.com/support/learning-center</a>
Support Knowledge Base	<a href="http://www.comsol.com/support/knowledgebase">www.comsol.com/support/knowledgebase</a>

# Overview of the User's Guide

The *Granular Flow Module User's Guide* gets you started with modeling granular flow using COMSOL Multiphysics. The information in this guide is specific to this module. Instructions how to use COMSOL in general are included with the *COMSOL Multiphysics Reference Manual*.




As detailed in the section [Where Do I Access the Documentation and Application Libraries?](#) this information can also be searched from the COMSOL Multiphysics software **Help** menu.

---

## MODELING GRANULAR FLOW

The [Modeling Granular Flow](#) chapter provides an overview of the physics interface and features available for modeling granular flow.

## THE GRANULAR FLOW INTERFACE

The [Granular Flow Interface](#) chapter describes the Granular Flow interface found under the **Fluid Flow** branch () when adding a physics interface. It also includes some theory for granular flow.



# Modeling Granular Flow

This chapter gives an overview of the physics interface and features available for modeling granular flow in COMSOL Multiphysics®.

In this chapter:

- [Introduction to Granular Flow Modeling](#)
- [General Tools for Granular Flow Modeling](#)

# Introduction to Granular Flow Modeling

Granular flow provides a Lagrangian description of a problem by solving ordinary differential equations using Newton's law of motion. Granular flow implements the discrete element method (DEM), which is a particle-based method that takes into account the translational and rotational degrees of freedom of the particles, which are referred to as *grains* in granular flow. DEM tracks the motion of individual grains by taking into account the forces on the grains due to external fields such as gravity and contact with other grains and walls to predict the bulk motion.

The trajectories of individual grains are always solved for in the time domain. The algorithms in the Granular Flow Module treat the grains as soft particles that can undergo elastic deformation during contact. The grain shape is spherical in 3D and is cylindrical in 2D. At each time step taken by the solver, the forces acting on each grain are queried from the external fields at the current grain position. The grain–grain and grain–wall collisions are detected, and contact force models are used to evaluate the forces due to contacts that are added to the total force on the grains. The grain degrees of freedom are then updated, and the process repeats until the specified end time for the simulation is reached.

Various contact force models are available in the Granular Flow Module, including both linear and nonlinear viscoelastic models. Additionally, noncontact force such as van der Waals force can also be included to take into account the long-range interactions on the grains. These models can also take into account the resistance to the rotational motion of the grains that resist rolling and twisting motion during contact.

Heat transfer effects on the grains can also be included by tracking the temperature of each grain. A grain's temperature can change due to an external heat source, convective heat transfer with the surroundings, and conductive heat transfer due to grain–grain and grain–wall contacts.

# General Tools for Granular Flow Modeling

In this section:

- [Special Variables](#)
- [Nonlocal Couplings](#)
- [Sampling from Random Number Distributions](#)
- [Study Setup](#)
- [The Grain Dataset](#)

## *Special Variables*

---

The Granular Flow interface defines a number of special variables, some of which can only be used during results processing. These variables can be found in the **Grain statistics** section of variables when you click **Insert Expression** or **Replace Expression** during results processing.

All of the variables described in this section are preceded by the physics interface identifier, typically `gran`. If multiple instances of the physics interface exist, the additional instances are followed by a number, for example, `gran` and `gran2`.

The following variables are defined:

- **Grain index**, `pidx`: Each grain is assigned a unique index starting from 1 up to the total number of grains. This expression can be passed into a function, which can create, for example, random forces that are unique for each grain. Suppose a random function has already been defined with name `rn1`, which takes 2 input arguments. Then a random force can be constructed with the expression `rn1(pidx, t)`.
- **Grain release feature**, `grf`: If there are multiple release features in a model, it is useful to be able to visualize how the grains mix together based on their initial release position. The **Grain release feature** variable takes a numeric value, starting at 1, which is unique to each release feature.



See *Mixing Grains in a Ribbon Mixer*: Application Library path `Granular_Flow_Module/Mixing_and_Separation/ribbon_mixer`.

---

- **Release time**, `rti`: Since grains can be released at arbitrary points in time, it is often useful to visualize at what time a specific grain was released.
- **Stop time**, `st`: The time at which a grain disappears at a boundary.
- **Final status**, `fs`: The final status of the grain. This indicates the status of a grain at a given point in time. When used during results processing, the value always indicates the status of the grain at the last time step. The value is an integer, which has one of the following values:
  - 0 for unreleased grains
  - 1 for grains that are still in the modeling domain
  - 4 for grains that have disappeared
- **Number of grains in contact**, `Ng`: The number of grains each grain is in contact with.
- **Number of wall elements in contact**, `Nw`: The number of wall elements each grain is in contact with. Contact is defined as any interaction that can induce a normal force on the grain and can therefore include interactions where the grain is not in physical contact with another grain or a wall element.
- The status of the grain: `particlestatus` (with no scope). This indicates the current status of each grain. The value is an integer with the same meaning as the final status of the grain. The variable `particlestatus` exists while computing grain trajectories, so it can be used in weak expressions defined on the grains but cannot be used during results processing.



The following variables are global and can therefore be evaluated using the **Global Evaluation** node under **Derived Values**. They do not have unique values for each grain.

---

The following variables are found in the **Grain statistics** section of variables when you click **Insert Expression** or **Replace Expression** during results processing:

- **Total number of grains released by feature**, `<tag>.Ntf`, where `<tag>` is the tag of a grain release feature, such as the [Release](#), [Inlet](#), [Release from Grid](#), or [Release from Data File](#) feature. This global variable is uniquely defined for each release feature, and gives the total number of grains that are successfully released by that feature.

### *Nonlocal Couplings*

---

The purpose of a model is often to compute the sum, average, maximum value, or minimum value of a quantity over a group of grains, such as the average kinetic energy

or the maximum residence time. A Granular Flow interface with the name `<phys>` (usually `gran`) creates the following operators:

- `<phys>.sum(expr)` evaluates the sum of the expression `expr` over the grains. The sum includes all grains that are active. It excludes grains that have not yet been released and those that have disappeared.
- `<phys>.sum_all(expr)` evaluates the sum of the expression `expr` over all grains, including grains that are not yet released or have disappeared. Since the coordinates of unreleased and disappeared grains are not-a-number (NaN), the sum may return NaN if the model includes unreleased or disappeared grains. An expression such as `pt.sum_all(isnan(qx))` can be used to compute the total number of unreleased and disappeared grains.
- `<phys>.ave(expr)` evaluates the average of the expression `expr` over the active grains. Unreleased and disappeared grains contribute to neither the numerator nor the denominator of the arithmetic mean.
- `<phys>.ave_all(expr)` evaluates the average of the expression `expr` over all grains. It is likely to return NaN if the model includes unreleased or disappeared grains.
- `<phys>.max(expr)` evaluates the maximum value of the expression `expr` over all active grains.
- `<phys>.max_all(expr)` evaluates the maximum value of the expression `expr` over all grains.

The treatment of NaN values in nonlocal maximum couplings can be platform-dependent, so use caution when evaluating the maximum over all grains including disappeared and unreleased grains.

- `<phys>.min(expr)` evaluates the minimum value of the expression `expr` over the active grains.
- `<phys>.min_all(expr)` evaluates the minimum value of the expression `expr` over all grains.

The treatment of NaN values in nonlocal minimum couplings can be platform-dependent, so use caution when evaluating the minimum over all grains including disappeared and unreleased grains.

- `<phys>.max(expr, evalExpr)` evaluates the expression `evalExpr` for the grain that has the maximum value of the expression `expr` out of all active grains. For example, in a model that uses the Granular Flow interface with the name `gran`, the expression `gran.max(gran.V, qx)` would evaluate the  $x$ -coordinate `qx` of the grain with the greatest velocity magnitude `gran.V`.

- `<phys>.max_all(expr, evalExpr)` evaluates the expression `evalExpr` for the grain that has the maximum value of the expression `expr` for all grains, including disappeared and unreleased grains.
- `<phys>.min(expr, evalExpr)` evaluates the expression `evalExpr` for the grain that has the minimum value of the expression `expr` out of all active grains. For example, in a model that uses the Granular Flow interface with name `gran`, the expression `gran.min(gran.V, qx)` would evaluate the  $x$ -coordinate `qx` of the grain with the smallest velocity magnitude `gran.V`.
- `<phys>.min_all(expr, evalExpr)` evaluates the expression `evalExpr` for the grain that has the minimum value of the expression `expr` for all grains, including disappeared and unreleased grains.

An instance of the Granular Flow interface with the default name `gran` defines the built-in nonlocal couplings shown in [Table 2-1](#).

TABLE 2-1: BUILT-IN NONLOCAL COUPLINGS FOR THE GRANULAR FLOW INTERFACE.

COUPLING NAME	DESCRIPTION
<code>gran.sum(a)</code>	Sum over grains
<code>gran.sum_all(a)</code>	Sum over all grains
<code>gran.ave(a)</code>	Average over grains
<code>gran.ave_all(a)</code>	Average over all grains
<code>gran.max(a)</code>	Maximum over grains
<code>gran.max_all(a)</code>	Maximum over all grains
<code>gran.min(a)</code>	Minimum over grains
<code>gran.min_all(a)</code>	Minimum over all grains
<code>gran.max(a,b)</code>	At maximum over grains
<code>gran.max_all(a,b)</code>	At maximum over all grains
<code>gran.min(a,b)</code>	At minimum over grains
<code>gran.min_all(a,b)</code>	At minimum over all grains

### *Sampling from Random Number Distributions*

Certain physics features in the Granular Flow interface such as the grain features utilize internal random number generators to sample the grain positions, release times, distributions of grain types, and initial values of auxiliary dependent variables. To be more specific, most of the time the above features depend on pseudorandom number generation (PRNG) rather than truly random number generation (RNG). The key

difference is that RNG is based on a real source of entropy, such as radioactive decay, whereas in PRNG the numbers are obtained according to an algorithm.

### **PSEUDORANDOM NUMBER GENERATION**

Even though pseudorandomly generated numbers are not truly random, in the sense that an observer with a perfect knowledge of the algorithm could predict the next number in a sequence given the previous numbers, from a practical standpoint PRNG is usually sufficient.

The pseudorandom numbers used in the Granular Flow Module are generally obtained from an internal implementation of a PRNG that is instantiated using a seed value. Controlling this seed can directly influence the behavior of the PRNG and hence the reproducibility of the model. This seed can be controlled by the **Seeds for random number generation** list in the physics interface **Advanced Settings** section.

The following options are available:

- **Unique** is the default option. When this option is selected, the seed is set to a predetermined value internally.
- When **Random** is selected, the seed is itself randomly generated every time the study is run. This will ensure that the solution is not reproducible when running a study multiple times.
- When **User defined** is selected, additional text fields appear in the settings windows for all nodes that use random numbers. The seed value can be provided directly by the user. A set of distinct solutions can be obtained by running a **Parametric Sweep** over several values of this argument.

For simple models, the **Unique** and **User defined** options may make the solution reproducible when rerunning the study multiple times. Reproducibility is somewhat easier to achieve when using a manual time step size. However, the results may not be 100% reproducible because even the slightest change in the time step size, down to machine precision, will cause different pseudorandom numbers to be generated. This can have a snowballing effect where the different solution values cause subsequent time steps to take on different sizes, leading to different pseudorandom numbers in all ensuing time steps. The **Random** option is never expected to make the solution reproducible across multiple runs of the study.

## *Study Setup*

---

For the Granular Flow interface, the grain trajectories must be computed using a **Time-Dependent Solver**. When modeling granular flow, it is common to utilize the results from one study step as the initial conditions for a subsequent study step with different physics features. For example, it is common to release grains and let them settle into a container before letting them flow or mix due to wall movements. This can be achieved by selecting the appropriate options in the [Values of Dependent Variables](#) section.



See *Powder Spreading Using a Moving Roller*, Application Library path **Granular\_Flow\_Module/Transport/powder\_spreading**.

---

### **SOLVER CONFIGURATIONS**

The Granular Flow interface only supports explicit time solver methods. The default time-stepping method is the second-order **Classical Runge–Kutta** method.



The default time step provided should be considered as a suggestion and you should carefully select the appropriate time step based on your model.

## *The Grain Dataset*

---

A special **Grain** dataset, selected from the **More Datasets** submenu, provides the possibility to visualize grain positions computed by a Granular Flow interface. The Grain dataset is automatically created when solving a model containing a Granular Flow interface, if the **Generate default plots** option is selected in the Study. Selections can be added to the grain dataset, which makes it possible to compute, for example, the number or fraction of grains in a given domain or on a given boundary during results processing. You can visualize the grains using a plot of the grain positions in the original dimension, as a Poincaré map, or as a 2D phase portrait.



## The Granular Flow Interface

This chapter describes the **Granular Flow (gran)** () interface found under the **Fluid Flow** branch () when adding a physics interface.

In this chapter:

- [The Granular Flow Interface](#)
- [Theory for the Granular Flow Interface](#)

# The Granular Flow Interface

The **Granular Flow (gran)** interface () , found under the **Fluid Flow** branch () when adding a physics interface, computes the contact forces in between grains and between grains and geometry walls. The grain motion is usually driven by external fields and is determined by Newton's second law.

When this physics interface is added, the following default nodes are also added to the **Model Builder**: [Grain Properties](#), [Wall](#), [Contact Between Grains](#), [Contact with Walls](#), and [Gravity](#). From the **Physics** toolbar, you can add other nodes that implement, for example, grain release features and additional external forces or torques. You can also right-click the **Granular Flow** to select physics features from the context menu.

The **Label** is the physics interface name. The default is `Granular Flow`.

The **Name** is used primarily as a scope prefix for variables defined by the physics interface. Refer to such physics interface variables in expressions using the pattern `<name>.<variable_name>`. In order to distinguish between variables belonging to different physics interfaces, the `name` string must be unique. Only letters, numbers, and underscores (`_`) are permitted in the **Name** field. The first character must be a letter.

The default **Name** (for the first physics interface in the model) is `gran`.

## FORCE

The Granular Flow interface uses contact force models to evaluate the contact forces as a function of displacement and velocity.

### *Contact Force Model*

Select an option from the **Contact force model** list: **Linear elastic**, **Hertz–MD** (the default) or **Hertz–MD with adhesion**.

- The **Linear elastic** model is a linear viscoelastic model that includes an elastic force that depends linearly on the overlap, and a viscous force. Spring constants can be specified in the Settings window for [Contact Between Grains](#) and [Contact with Walls](#) nodes.
- The **Hertz–MD** model is a nonlinear viscoelastic model that includes a more realistic nonlinear force–displacement relationship.
- The **Hertz–MD with adhesion** model is an extension of the **Hertz–MD** model that is used to model adhesive forces during grain–grain and grain–wall interactions. These

adhesive forces are tensile forces that arise due to surface adhesion and can cause a prolonged contact after collision.

#### *Compute van der Waals Force*

When **Hertz–MD** or **Hertz–MD with adhesion** model is selected, the **Compute van der Waals force** checkbox appears. This checkbox is not selected by default. Selecting this checkbox will add a van der Waals force to the grains, in addition to the forces computed by the **Contact force model**.



For theory see:

- [Contact Force: Linear Elastic Model](#)
  - [Contact Force: Hertz–MD \(Mindlin and Deresiewicz\) Model](#)
  - [Contact Force: Hertz–MD with Adhesion Model](#)
  - [Contact Force: van der Waals Force](#)
- 

### **ROTATIONAL RESISTANCE**

Select an option from the **Rotational resistance model** list: **Constant torque model**, **Varying torque model** (the default), or **None**. Friction coefficients can be specified using the Settings window for [Contact Between Grains](#) and [Contact with Walls](#) nodes.

- The **Constant torque model** applies a constant rotational resistant torque proportional to the normal contact force on colliding grains.
  - The **Varying torque model** applies a continuous rotational resistant torque that depends on the angular velocity of grain and is limited to the maximum of constant resistant torque discussed for the **Constant torque model**.
  - No rotational resistance torque is applied when **None** is selected.
- 



[Rotational Resistance Theory](#)

---

### **ADDITIONAL VARIABLES**

Use the settings in this section to add additional variables to the model that can affect the solution and provide additional information about the grains.

#### *Compute Grain Temperature*

Select the **Compute grain temperature** checkbox to compute grain temperatures. This checkbox is not selected by default. When this option is selected, the temperature of

the grain is computed by solving an additional ordinary differential equation per grain, which accounts for various heat sources including conduction, convection, and external heat sources. Thermal properties for the grains can be specified in the Settings window for the [Grain Properties](#) node. Initial value of grain temperature can be specified using the Settings window for any grain release features used.

#### *Compute Conductive Heat Transfer*

When the **Compute grain temperature** checkbox is selected, the **Compute conductive heat transfer** checkbox is available when the contact force model is either **Hertz–MD** or **Hertz–MD with adhesion**. This checkbox is not selected by default. Selecting this checkbox includes the conductive heat source in the grain–grain and grain–wall heat transfer calculations. Thermal properties for the walls can be specified in the Settings window for the [Wall](#) node.



## Computing Grain Temperature

---

#### *Compute Coordination Numbers*

The **Compute coordination numbers** checkbox is not selected by default. When selected, two variables are added to each grain to compute their coordination numbers. The two variables that are added are:

- The number of grains each grain is in contact with (variable name `<name>.Ng`).
- The number of wall elements each grain is in contact with (variable name `<name>.Nw`).

Contact is defined as any interaction that can induce a normal force on the grain, and it can therefore include interactions where the grain is not in physical contact with another grain or a wall element.

#### *Out-of-Plane Grain Thickness*


The **Out-of-plane grain thickness** is only available in 2D components. Enter a value for the **Out-of-plane grain thickness** (SI unit: m). The default is 1 m. In [The Granular Flow Interface](#), 2D components are treated as a planar representation of a 3D component

with cylindrical grains and planar walls in the out-of-plane direction. The **Out-of-plane grain thickness** denotes the extent of the 3D component in that direction.



Depending on the various combinations of the **Contact force model**, **Rotational resistance model**, **Compute grain temperature**, and **Compute conductive heat transfer** checkboxes that are selected, different settings can appear in the Settings windows of some of the default nodes: [Grain Properties](#), [Wall](#), [Contact Between Grains](#), [Contact with Walls](#), and nodes used for releasing the grains. The available options for the different selections are explained in the respective features.

### ADVANCED SETTINGS

This section is only shown when **Advanced Physics Options** are enabled (click the **Show More Options** button (  ) on the **Model Builder** toolbar, and select **Advanced Physics Options** in the **Show More Options** dialog).

#### *Seeds for Random Number Generation*

Select an option from the **Seeds for random number generation** list: **Unique** (the default), **Random**, or **User defined**.

Many release features in the [The Granular Flow Interface](#) utilize pseudorandom number generators (PRNGs) to sample the grain positions, release times, distribution of grain properties and initial values of auxiliary dependent variables. The seed for these internal PRNGs are controlled by this setting.

- When **Unique** is selected, the seeds are set automatically to a unique value.
- When **Random** is selected, the seeds are set automatically to a random value that depends on machine time. This will ensure that the solution is not reproducible when running a study multiple times.
- When **User defined** is selected, additional text fields appear in the settings windows for all nodes that use random numbers. This number is used as the seed value. A set of distinct solutions can be obtained by running a **Parametric Sweep** over several values of this argument.

Note that these PRNGs produce pseudorandom numbers and not truly random numbers derived from a natural entropy source. For simple models, the **Unique** and **User defined** options may make the solution reproducible when rerunning the study multiple times. Reproducibility is somewhat easier to achieve when using a manual time step size. However, the results may not be 100% reproducible because even the

slightest change in the time step size, down to machine precision, will cause different pseudorandom numbers to be generated. This can have a snowballing effect where the different solution values cause subsequent time steps to take on different sizes, leading to different pseudorandom numbers in all ensuing time steps. The **Random** option is never expected to make the solution reproducible across multiple runs of the study.

#### *Maximum Number of Cells per Direction*

The broad search step in the [Contact Search Theory](#) algorithm for detecting grain–grain contact includes the construction of a grid across the modeling domains. The default size of the grid cells is controlled by the contact search radius, which can sometimes lead to large memory requirements, especially when the ratio of grain diameters to the geometry dimensions are very small. This can be mitigated by controlling the maximum number of grid cells in each direction.

Enter the values of the **Maximum number of cells per direction** to control the size of the grid in each direction ( $x, y$  in 2D and  $x, y, z$  in 3D) independently. The default values for each direction is 1000.

#### **DEPENDENT VARIABLES**

The dependent variables (field variables) are the **Grain center position**, **Grain center position components**, **Grain velocity**, and **Grain velocity components**.



### Theory for the Granular Flow Interface

---

#### *List of Granular Flow Interface Physics Features*

---

The Granular Flow interface has the following nodes and subnodes (listed in alphabetical order) available from the **Physics** ribbon toolbar (Windows users), **Physics** context menu (macOS or Linux users), or right-click to access the context menu (all users).



In general, to add a node, go to the **Physics** toolbar, no matter what operating system you are using. Subnodes are available by clicking the parent node and selecting it from the **Attributes** menu.

---

- Accumulator (Boundary)
- Accumulator (Domain)
- Auxiliary Dependent Variable
- Bounding Box
- Contact Between Grains
- Contact with Walls
- Convective Heat Transfer
- Force
- Force Accumulator
- Grain Counter
- Grain Properties
- Gravity
- Heat Source
- Inlet
- Outlet
- Periodic Condition
- Release
- Release from Data File
- Release from Grid
- Torque
- Wall

### *Grain Properties*

---

Use the **Grain Properties** node to specify the grain material properties and grain size.

#### **GRANULAR MATERIAL PROPERTIES**

You can either type values or expressions directly or take them from a **Material** node, which could either be a user-defined **Blank Material** or a material from one of the material libraries.

The default value in the **Granular material** list is **None**. If you want to use data from a **Blank Material** or from the material libraries, first add the material to the model (right-click **Materials** either under the model component or under **Global Definitions**, and then select it from the list).

For the **Density**  $\rho_g$  (SI unit:  $\text{kg}/\text{m}^3$ ), by default this is taken **From material**. If **User defined** is selected, the default is  $2200 \text{ kg}/\text{m}^3$ .

When a **Hertz–MD** or **Hertz–MD with adhesion** model is selected in the **Contact force model** list in the physics interface **Force** section, the **Specify** list appears with the options

- **Young’s modulus and Poisson’s Ratio** (the default)
- **Young’s modulus and shear modulus**

When **Young’s modulus and Poisson’s ratio** is selected, **Young’s modulus** and **Poisson’s ratio** fields appear; and when **Young’s modulus and shear modulus** is selected, **Young’s**

**modulus** and **Shear modulus** fields appear. These properties can be assigned to grains as follows:

- **Young's modulus**  $E_g$  (SI unit: Pa). By default this is taken **From material**. If **User defined** is selected, the default is 100 GPa.
- **Shear modulus**  $G_g$  (SI unit: Pa). By default this is taken **From material**. If **User defined** is selected, the default is 45.45 GPa.
- **Poisson's ratio**  $\nu_g$ . By default this is taken **From material**. If **User defined** is selected, the default is 0.1.

The other material properties that can be assigned to grains are as follows:

- **Specific heat capacity**  $C_{p,g}$  (SI unit: J/(kg.K)). By default this is taken **From material**. If **User defined** is selected, the default is 2000 J/(kg.K). This field is only available when **Compute grain temperature** checkbox is selected in the **Grain Temperature** section in the physics interface.
- **Thermal conductivity**  $k_g$  (SI unit: W/(m.K)). By default this is taken **From material**. If **User defined** is selected, the default is 0.2 W/(m.K). This field is only available when the **Compute conductive heat transfer** checkbox is selected in the **Grain Temperature** section in the physics interface.

## SIZE

Enter value or expression for the **Grain diameter**  $d_g$  (SI unit: m). The default value is 1 mm.

Enter value or expression for the **Contact search expansion ratio**  $\beta$ .  $\beta$  is defined as the ratio of the contact search diameter to the grain diameter. The default value is 1 which means the contact search diameter equals the grain diameter. This value must always be greater than or equal to 1 and can be used to extend the search radius used during contact detection. Typical use cases include enlarging the search radius to account for noncontact forces in force models such as **Hertz–MD with adhesion** or **van der Waals**.

## ADHESION PROPERTIES

Enter value or expression for the **Surface energy density**  $\gamma$  (SI unit: J/m<sup>2</sup>). The default value is 0.05. This field is only available when the **Compute van der Waals force**

checkbox is selected and/or when the **Hertz–MD with adhesion** is selected in the **Contact force model** list in the physics interface **Force** section.



The **Grain Properties** node is a default feature and cannot be deleted or disabled, and it is not possible to paste or duplicate this node in the model builder.

However, it is possible to add more than one instance of this **Grain Properties** node to the same model. When you do so, each instance corresponds to a different species of grain. Unlike the default **Grain Properties** node, these nondefault nodes can be deleted, disabled, duplicated, or pasted in the Model Builder.

---

## *Wall*

---

Use a **Wall** node to specify a wall's material properties and movement. The [Accumulator \(Boundary\)](#) and [Force Accumulator](#) subnodes are available from the context menu (right-click the parent node) or from the **Physics** toolbar, **Attributes** menu.

### **WALL MATERIAL PROPERTIES**

When **Hertz–MD** or **Hertz–MD with adhesion** model is selected in the [Contact Force Model](#) list in the physics interface **Force** section, the **Specify** list is available with the options

- **Young's modulus and Poisson's ratio** (the default)
- **Young's modulus and shear modulus**

**Young's modulus** and **Poisson's ratio** fields appear when **Young's modulus and Poisson's ratio** is selected. **Young's modulus** and **Shear modulus** fields appear when **Young's modulus and shear modulus** is selected. These properties can be assigned to grains as follows:

- **Young's modulus**  $E_g$  (SI unit: Pa). By default this is taken **From material**. If **User defined** is selected, the default is 100 GPa.
- **Shear modulus**  $G_g$  (SI unit: Pa). By default this is taken **From material**. If **User defined** is selected, the default is 45.45 GPa.
- **Poisson's ratio**  $\nu_g$ . By default this is taken **From material**. If **User defined** is selected, the default is 0.1.

## BOUNDARY CONDITION

This section is only available when **Compute conductive heat transfer** checkbox is selected in the physics interface **Grain Temperature** section. Select a **Specify boundary condition**:

- **Specified temperature** (default) — enter a value or expression for the **Temperature  $T$**  (SI unit: K) which defines the temperature of selected walls. The default value is 293.15 K. If another physics interface is present which computes the temperature of the walls then this can be selected from the list.
- **Insulated** — select this when you want to impose insulated boundary condition on the walls.

## ADHESION PROPERTIES

This section is only available when the **Compute van der Waals force** checkbox is selected and/or when the **Hertz–MD with adhesion** is selected in the **Contact force model** list in the physics interface **Force** section. Enter a value or expression for the **Surface energy density  $\gamma$**  (SI unit: J/m<sup>2</sup>). The default value is 0.05.

## WALL MOVEMENT

This section controls the rigid-body motion of the selected walls. Select an option from the **Wall motion** list.

- **Fixed** (default) — select this when you want the selected walls to be stationary.
- **Translation** — select this when you want to impose any arbitrary prescribed displacement to the walls. This is commonly used to specify translational motion of the wall but can also be used to prescribe any arbitrary displacement.
- **Rotation** — select this when you want to impose a rotational displacement to the walls.
- **Translation and rotation** — select this when you want to impose a mixture of translational and rotational motion to the walls.



It is not recommended to use the Moving Mesh interface with the Granular Flow interface to describe any kind of wall movement. Use of the Moving Mesh interface with the Granular Flow interface is not fully supported and may result in unphysical grain dynamics.

---

When either **Translation** or **Translation and rotation** is selected, enter the values of the components of the **Prescribed displacement**  $dx$  (SI unit: m). The default value for each component is 0 m.

When either **Rotation** or **Translation and rotation** is selected, select an option from the **Rotation type** list.

- Choose **Constant angular velocity** (the default) to specify an angular velocity  $\omega$  in the **Angular velocity** field (SI unit: rad/s) using only numbers and model parameters. The default value is 0. Specify an initial angle  $\alpha_0$  in the **Initial angle** field (SI unit: rad). The default value is 0. This effectively sets the rotation angle to  $\alpha = \alpha_0 + \omega t$ .
- Choose **General angular velocity** to specify a general angular velocity  $\omega$  using an expression in the **Angular velocity** field (SI unit: rad/s) with an initial angle  $\alpha_0$  specified in the **Initial angle** field (SI unit: rad). The rotation angle is computed by solving an ODE for  $\alpha$ , which is therefore a state variable and part of the solution.
- Choose **User defined** to add a user-defined expression for the rotation angle  $\alpha$  in the **Rotation angle** field. The default value is 0.

Enter the base point  $\mathbf{r}_{bp}$  components in the **Rotation axis base point** fields (SI unit: m). The default value for each component is 0.



When **Translation and rotation** is selected, the translational motion of the  $\mathbf{r}_{bp}$  components is automatically accounted for and should not be explicitly included in the expressions.

---

For 3D components, enter the rotation axis  $\mathbf{e}_{ax}$  in the **Rotation axis** fields. The default values are 0, 0, 1. In 2D, no rotation axis is specified; it is assumed to point out of the plane, toward the observer.



If you specify a **General angular velocity**, the **Wall** feature uses an ODE to integrate the rotation angle in time. This can in some situations negatively affect solver behavior. Therefore, when the rotational velocity is constant, make sure to select **Constant angular velocity**.

---

### *Contact Between Grains*

---

Use the **Contact Between Grains** node to define the contact pair properties between different species of grains.

## GRAINS PAIR SELECTION

Select **All pairs** (default) or **Manual** option from the **Selection** list.

- When **All pairs** is selected, the contact properties defined applies to all possible pairs between different species of grains.
- When **Manual** is selected, a table is available with two columns **Specify properties for first grain type** and **Specify properties for second grain type**. You can specify any pairs of grain species to have pair properties defined by this node. This table also has options to move rows up or down, add a new row, delete selected rows, and clear the entire table.



The **Selection** list is disabled in the default **Contact Between Grains** node. The default node always specifies contact pair properties for all possible pairs between grains. If you want to specify different contact pair properties for different grains pairs, you can add **Contact Between Grains** nodes. Then select **Manual** from the **Selection** list and edit the table with desired pairs between different species of grains.



For example, if you set up a model with two [Grain Properties](#) nodes to define two species of grains with labels **Grain Properties 1** and **Grain Properties 2**, there are three possible pairs between different grain species:

- **Grain Properties 1** and **Grain Properties 1**
- **Grain Properties 1** and **Grain Properties 2**
- **Grain Properties 2** and **Grain Properties 2**

Assume that the first two pairs in the above list have the same contact pair properties but the third pair has a different set of contact pair properties. The default **Contact Between Grains** node specifies properties for all three pairs. Now, if you add one more **Contact Between Grains** node and select **Manual** from the **Selection** list in the **Grains Pair Selection** section, then you can specify **Grain Properties 2** in **Specify properties for first grain type** and again **Grain Properties 2** in **Specify properties for second grain type**. Now, all the values entered in this node will only apply to this selected pair and the previous values assigned for this specific pair by the default node get overridden.

## CONTACT PROPERTIES

Enter values or expressions for the following quantities:

- **Normal coefficient of restitution**  $e_n$ . The default value is 1. This is the value of coefficient of restitution in normal direction or direction along the line connecting centers of two grains in contact. It determines the amount of energy loss in the normal direction when the grains come in contact.
- **Tangential coefficient of restitution**  $e_t$ . The default value is 1. This is the value of the coefficient of restitution in the tangential direction or direction perpendicular to the line connecting centers of two grains in contact. It determines the amount of energy loss in the tangential direction when the grains come in contact.
- **Normal spring constant**  $k_n$  (SI unit: N/m). The default value is 10 MN/m. This setting is only available when **Linear elastic** is selected from the **Contact force model** list in the physics interface node's **Force** section. The normal spring constant for other contact force models is automatically calculated based on the material properties and overlapping distance between two grains in contact.
- **Tangential spring constant**  $k_t$  (SI unit: N/m). The default value is 10 MN/m. This setting is only available when **Linear elastic** is selected from the **Contact force model** list in the physics interface **Force** section. The tangential spring constant for other contact force models is automatically calculated based on the material properties and overlapping distance between two grains in contact.
- **Static friction coefficient**  $\mu_n$ . The default value is 0.09.
- **Rolling friction coefficient**  $\mu_r$ . The default value is 0.1. This setting is only available when the selected **Rotational resistance model** is either **Constant torque model** or **Varying torque model** in the physics interface node's **Rotational Resistance** section.
- **Twisting friction coefficient**  $\mu_{tw}$ . The default value is 0.1. This setting is only available in 3D when the selected **Rotational resistance model** is either **Constant torque model** or **Varying torque model** in the physics interface node's **Rotational Resistance** section.

---

For theory see:



- [Contact Force: Linear Elastic Model](#)
  - [Contact Force: Hertz–MD \(Mindlin and Deresiewicz\) Model](#)
  - [Contact Force: Coulomb's Criterion](#)
  - [Rotational Resistance Theory](#)
-

## ADHESION PROPERTIES

This section is only available when

- the **Compute van der Waals force** checkbox is selected and/or
- **Hertz–MD with adhesion** is selected from the **Contact force model** list

in the physics interface node’s **Force** section. Enter values or expressions for the following quantities:

- **Interface energy density**  $\gamma_{ij}$ . The default value is 0.
- **Cutoff distance for van der Waals force**  $D_{\max}$  (SI unit: m), only available when the **Compute van der Waals force** checkbox is selected in the physics interface node’s **Force** section. The default value is 1  $\mu\text{m}$ .
- **Minimal distance for van der Waals force**  $D_{\min}$  (SI unit: m), only available when the **Compute van der Waals force** checkbox is selected in the physics interface node’s **Force** section. The default value is 1.65 angstrom.

---

For theory see:



- [Contact Force: Hertz–MD with Adhesion Model](#)
- [Contact Force: van der Waals Force](#)

---

## TEMPERATURE CORRECTION FACTOR

This section is only available when the **Compute conductive heat transfer** checkbox is selected in the physics interface node’s **Additional Variables** section.

Enter a value or expression for **Temperature correction factor for contact radius**  $C_r$ . The default is 1. This correction factor can be used to account for the large contact radius that often results from utilizing artificially low values of Young’s modulus.



[Contact Heat Transfer](#)

---

## *Contact with Walls*

---

Use the **Contact with Walls** node to define the contact pair properties between grains and walls.

## GRAIN–WALL PAIR SELECTION

Select **All pairs** (default), **Pair between all walls and selected grain types**, **Pair between all grains and selected wall types**, or **Manual** from the **Selection** list.

- When **All pairs** is selected, the contact properties defined applies to all possible pairs between grains and walls.
- When **Pair between all walls and selected grain types** is selected, a table is available with one column titled **Specify properties of grain** where you can add any species of grains. You can specify contact pair properties between selected grain species and all types of walls defined by different [Wall](#) nodes.
- When **Pair between all walls and selected wall types** is selected, a table is available with one column titled **Specify properties of wall** where you can add any types of walls. You can specify contact pair properties between selected type of wall and all species of grains defined by different [Grain Properties](#) nodes.
- When **Manual** is selected, a table is available with two columns: **Specify properties for wall** and **Specify properties for grain**. You can specify any pairs of grain species and wall types to have pair properties defined by this node.

All the tables mentioned above have options to move rows up or down, add a new row, delete selected rows, and clear the entire table.



The **Selection** list is disabled in the default **Contact with Walls** node. The default node always specifies contact pair properties for all possible pairs between grains and walls. If you want to specify different contact pair properties for different grain–wall pairs, you can add **Contact with Walls** nodes. Then select an appropriate option from the **Selection** list and edit the table with desired pairs between walls and grains.

---

---

For example, if you set up a model with two [Grain Properties](#) nodes to define two species of grains with labels **Grain Properties 1** and **Grain Properties 2**, and one [Wall](#) node labeled **Wall 1** to define a type of wall, then there are two possible pairs between grain and wall:

- **Wall 1** and **Grain Properties 1**
- **Wall 1** and **Grain Properties 2**

Assume that each pair has a different set of contact pair properties. The default **Contact with Walls** node specifies properties for both pairs. Now, if you want to specify different pair properties for the second pair in the above list, you can add a new **Contact with Walls** node and then specify the pair using one of the following two ways:



- Select the **Pair between all walls and selected grain type** option from the **Selection** list in the **Grain–Wall Pair Selection** section. Choose **Grain Properties 2** for the **Specify properties for grain** column in the table.
- Select the **Manual** option from the **Selection** list in the **Grain–Wall Pair Selection** section. Choose **Wall 1** for the **Specify properties for wall** column and **Grain Properties 2** for the **Specify properties for grain** column in the table.

Now all the values entered in this node will only apply to this selected pair, and the previous values assigned for this specific pair by the default node get overridden.

---

## CONTACT PROPERTIES

Enter a value or expressions for the following:

- **Normal coefficient of restitution**  $e_n$ . The default value is 1. This is the value of the coefficient of restitution in the normal direction or direction along the line connecting center of the grain and the point of contact at wall. It determines the amount of energy loss in the normal direction when the grain and wall come in contact.
- **Tangential coefficient of restitution**  $e_t$ . The default value is 1. This is the value of the coefficient of restitution in the tangential direction or direction perpendicular to the line connecting center of the grain and the point of contact at wall. It determines

the amount of energy loss in the tangential direction when the grains come in contact.

- **Normal spring constant**  $k_n$  (SI unit: N/m). The default value is 10 MN/m. This setting is only available when **Linear elastic** is selected from the **Contact force model** list in the physics interface node's **Force** section. The normal spring constant for other contact force models is automatically calculated based on the material properties and overlapping distance between grain and wall in contact.
- **Tangential spring constant**  $k_t$  (SI unit: N/m). The default value is 10 MN/m. This setting is only available when **Linear elastic** is selected from the **Contact force model** list in the physics interface node's **Force** section. The tangential spring constant for other contact force models is automatically calculated based on the material properties and overlapping distance between grain and wall in contact.
- **Static friction coefficient**  $\mu_n$ . The default value is 0.09.
- **Rolling friction coefficient**  $\mu_r$ . The default value is 0.1. This setting is only available when the option selected in **Rotational resistance model** is either **Constant torque model** or **Varying torque model** in the physics interface node's **Rotational Resistance** section
- **Twisting friction coefficient**  $\mu_{tw}$ . The default value is 0.1. This setting is only available in 3D when the option selected in **Rotational resistance model** is either **Constant torque model** or **Varying torque model** in the physics interface node's **Rotational Resistance** section.

---

For theory see:



- [Contact Force: Linear Elastic Model](#)
- [Contact Force: Hertz–MD \(Mindlin and Deresiewicz\) Model](#)
- [Contact Force: Coulomb's Criterion](#)
- [Rotational Resistance Theory](#)

---

## ADHESION PROPERTIES

This section is only available when

- the **Compute van der Waals force** checkbox is selected and/or
- **Hertz–MD with adhesion** is selected from the **Contact force model** list

in the physics interface node's **Force** section. Enter the value or expression for the following:

- **Interface energy density**  $\gamma_{ij}$ . The default value is 0.
- **Cutoff distance**  $D_{\max}$  (SI unit: m), only available when the **Compute van der Waals force** checkbox is selected in the physics interface node's **Force** section. The default value is 1  $\mu\text{m}$ .
- **Minimal distance**  $D_{\min}$  (SI unit: m), only available when the **Compute van der Waals force** checkbox is selected in the physics interface node's **Force** section. The default value is 1.65 angstrom.



For theory see:

- [Contact Force: Hertz–MD with Adhesion Model](#)
- [Contact Force: van der Waals Force](#)

---

#### TEMPERATURE CORRECTION FACTOR

This section is only available when the **Compute grain temperature** and **Compute conductive heat transfer** checkboxes are selected in the physics interface node's **Additional Variables** section.

Enter the value or expression for the **Temperature correction factor for contact radius**  $C_r$ . The default is 1. This correction factor can be used to account for the large contact radius that often results from utilizing artificially low values of Young's modulus.



[Contact Heat Transfer](#)

---

#### *Gravity*

The **Gravity** node is a default node and can be used to exert a gravitational force on grains. The gravity vector can point in any direction with any magnitude, although the default is for grains to move downward using the acceleration due to gravity at Earth's surface. This feature can only be added once to the model and it applies to all grains throughout the model.

## ACCELERATION OF GRAVITY

Enter coordinates based on space dimension for the **Acceleration of gravity  $g$**  (SI unit:  $\text{m/s}^2$ ). The default magnitude for the gravity vector is  $g\_const$ , which is a built-in physical constant equal to  $g = 9.80665 \text{ m/s}^2$  corresponding to the standard acceleration due to gravity on Earth. The default direction is the negative  $y$  direction in a 2D geometry or the negative  $z$  direction for a 3D geometry.

### *Release*

---

Use the **Release** node to release grains into the model from a selected set of domains. The release times, initial values of the grains' degrees of freedom such as positions, velocities and temperature, and initial value of any auxiliary dependent variables can be specified.


The grains are released sequentially at each release time. For each grain, a release position and radius are first determined based on the options provided in the **Initial Position** and the **Released Grain Properties** sections. An attempt is made to release the grain at this position, and the attempt is considered successful only if the new grain does not overlap with any existing grains or wall elements. Alternatively, a maximum amount of allowed overlap can also be specified.

If the attempt is deemed successful, the grain is released at the release position. Alternatively, an unsuccessful attempt can be followed by a number of attempts at nearby release positions. Once the maximum number of attempts is reached, the grain is discarded, and the next grain is attempted.

## RELEASE TIMES

Select a **Distribution function**: **List of values** (default), **Uniform**, **Normal**, or **Lognormal**.

### *List of Values*

Enter **Release times** (SI unit: s), or click the **Range** button () to select and define a range of specific times. At each release time, grains are released with initial position and velocity as defined in the following sections.

### *Uniform*

Enter the **Number of values**, along with the **First time value** (SI unit: s) and the **Last time value** (SI unit: s). In addition, select whether the **Sampling from distribution** should be **Deterministic** or **Random**. When **Deterministic** is selected, an array of length **Number of values** of uniformly spaced release times is generated. This array starts with the **First time value** and ends with the **Last time value** exactly. The release times are reproducible each time the solution is computed.

When **Random** is selected, an array of random numbers of length **Number of values**, with a minimum lower limit of the **First time value** and maximum upper limit of the **Last time value** is generated. In this case, a release time of exactly the first and last time values is extremely unlikely.

#### *Normal and Lognormal*

Enter the **Number of values** along with the **Mean** (SI unit: s) and the **Standard deviation** (SI unit: s). In addition, select whether the **Sampling from distribution** should be **Deterministic** or **Random**. When **Deterministic** is selected, an ordered array of length **Number of values** of normally distributed release times is generated. The release times are more closely spaced around the value entered for the **Mean** and the spacing of the release time drops off according to the value of the **Standard deviation**. The release times are reproducible each time the solution is computed.

When **Random** is selected, the normal or lognormal distribution is generated by pseudorandom sampling, so there may be some statistical error.

## **INITIAL VALUES**

### *Position*

Select a **Position** from the list: **Random** (the default) or **Density**.

### **Random**

For **Random** the grains are released at random positions within the selected entities. If grains are released at multiple release times, the initial positions are uniquely generated for each release time. By contrast, for **Density** the grains positions are the same at each release time.

The grain positions are determined in the following way. First a random mesh element is selected for each grain with a probability proportional to the element size, so that grains are more likely to be released from larger mesh elements than smaller elements. After the mesh element is selected, random local coordinates are chosen within the element, then converted to global coordinates. A grain is then attempted to be placed at this location.

### **Density**

For **Density** the grains are positioned in the selected domains by sampling from a user-defined spatial distribution. Enter a value or expression for the **Density proportional to**  $\rho$  (dimensionless). The default is 1.

The **Density proportional to** field is an expression; the resulting grain distribution approximately has a density that is proportional to this expression. The resulting distribution looks a bit random, and it depends on the underlying mesh.

Select a **Release distribution accuracy order** between 1 and 5 (the default is 5), which determines the integration order that is used when computing the number of grains to release within each mesh element. The higher the accuracy order, the more accurately grains will be distributed among the mesh elements.

The **Position refinement factor** (default 0) must be a nonnegative integer. When the refinement factor is 0, each grain is always assigned a unique position, but the density is taken as a uniform value over each mesh element. If the refinement factor is a positive integer, the distribution of grains within each mesh element is weighted according to the density. Further increasing the **Position refinement factor** increases the number of evaluation points within each mesh element.

#### *Velocity*

Enter values or expressions for the components of the initial grain velocity  $\mathbf{v}_0$  (SI unit: m/s) based on space dimension. The defaults are 0 m/s.

#### *Temperature*

This section is only shown when the **Compute grain temperature** checkbox is selected in the physics interface **Grain Temperature** section. Enter a value or expression for the initial grain temperature  $T_{g,0}$  (SI unit: K). The default value is 293.15 K.

#### *Angular Velocity in body-fixed frame*

Enter values or expressions for the components of the initial angular velocity in the body-fixed frame  $\omega_0$  (SI unit: rad/s).

### **RELEASED GRAIN PROPERTIES**

This section is used to specify the total number of grains and the distribution of the grain properties to be released per release time. Select a **Distribution of released grain properties: Number of grains** (default), **Number fraction**, **Mass fraction**, or **Volume fraction**.

- When **Number of grains** is selected, enter the number of grains to be released for each grain type corresponding to a **Grain Properties** node. The default value for the first row in the table is 1, and it is 0 for all other rows.
- When one of the **Number fraction**, **Mass fraction**, or **Volume fraction** options is selected, enter the **Number of grains per release** (default 1). Then enter the values of the corresponding fractions of the distribution for each grain type. The default value

for the first row in the table is 1, and it is 0 for all other rows. The fractional values are normalized internally so that their sum is 1 even if the provided values do not add up to 1.

The values of the distribution can be numerical values or parameters defined in the **Parameters** node.

---

The grains are released sequentially, and each grain is associated with a **Grain Properties** node with a probability dictated by the user-specified distributions.



Further, the grains are only released if the overlap with existing grains and wall elements is acceptable. Therefore, the total number of grains being released (and the distribution of their properties) at release time should be viewed as maximum values (ideal distributions), and may not always exactly match the user-specified values. The number of grains being released and their distributions may therefore also be different for each release time specified in the **Release Times** section.

The user should therefore test the released grain populations to ensure an appropriate population is attained. Furthermore, for performance reasons, it is advised to ensure that the **Number of grains per release** is set as close to the actual number of grains being released as possible.



The algorithms used for the contact detection are described in [Contact Force: Linear Elastic Model](#) section in the [Theory for the Granular Flow Interface](#).

---

#### INITIAL VALUE OF AUXILIARY DEPENDENT VARIABLES

This section is available if an [Auxiliary Dependent Variable](#) has been added to the model. For each of the **Auxiliary Dependent Variables** (rp) added to the model, enter an initial value (rp<sub>0</sub>). The default value is 0.

#### ADVANCED SETTINGS

Enter a nonnegative value or expression for the **Maximum allowed normal overlap** (SI unit: m/s). The default is 0 m. This value is used to control the amount of overlap that is acceptable between a grain being considered for release and existing grains or wall elements.

Enter a value or expression for the **Number of release attempts per grain** (default 10). This value controls the number of release positions that are attempted for each grain.

If **User Defined** is selected from the **Seeds for random number generator** list in the physics interface **Advanced Settings** section, the **Seed for random number generator** text field is available. Enter the seed value of the pseudorandom number generator (PRNG) used by this feature. The default value is 1. The PRNG is used to generate random numbers for sampling grain release times, positions, grain properties, and initial values of auxiliary dependent variables.

### *Inlet*

---

Use the **Inlet** node to release grains into the modeling domain from selected boundaries. The released grains are positioned such that their centers lie on the selected boundaries. The selected boundaries are not included in the grain–wall interactions.

See [Release](#) for information about the following sections: **Release Times**, **Released Grain Properties**, **Initial Value of Auxiliary Dependent Variables**, and **Advanced Settings**.

### **INITIAL VALUES**

See the [Release](#) node for information about **Velocity**, **Angular velocity in body-fixed frame** and **Temperature**.

#### *Position*

Select an option from the **Position** list: **Random** (the default), **Density**, or **Uniform distribution** (2D components), or **Projected plane grid** (3D components). [Density](#), and [Random](#) have the same settings as described for the [Release](#) node.

- For 2D components, if **Uniform distribution** is selected, the union of the selected edges is divided into  $N$  segments of approximately equal length, and a grain is placed in the middle of each segment. Here,  $N$  refers to the total number of grains being attempted per release time.
- For 3D components, if **Projected plane grid** is selected, the grains are distributed on a plane grid in planes that are approximately tangential to the selected boundaries.

### *Release from Grid*

---

Use the **Release from Grid** node to release grains from a grid of initial positions with user-defined coordinates.

See [Release](#) for information about the following sections: **Release Times** and **Initial Value of Auxiliary Dependent Variables**.


## INITIAL VALUES

See the [Release](#) node for information about **Velocity**, **Angular velocity in body-fixed frame**, and **Temperature**.

### *Position*

Select an option from the **Position grid type** list: **All combinations** (the default) or **Specified combinations**.

### *Linear, Rectangular, and Arbitrary Grids*

For **All combinations** and **Specified combinations**, enter **Initial coordinates** based on space dimension ( $q_{x,0}$ ,  $q_{y,0}$ , and  $q_{z,0}$  for 3D components) for the grain positions, or click the **Range** button (  ) to select and define a range of specific coordinates.

If **Specified combinations** is selected, the number of initial coordinates entered for each space dimension must be equal, and the total number of grains released is equal to the length of one of the lists of initial coordinates. If **All combinations** is selected, the total number of grains released is equal to the product of the lengths of each list of initial coordinates.



For example, suppose a 2D model component includes a **Release from Grid** node with the following initial coordinates:

- $q_{x,0} = \text{range}(0,1,3)$
- $q_{y,0} = \text{range}(2,2,8)$

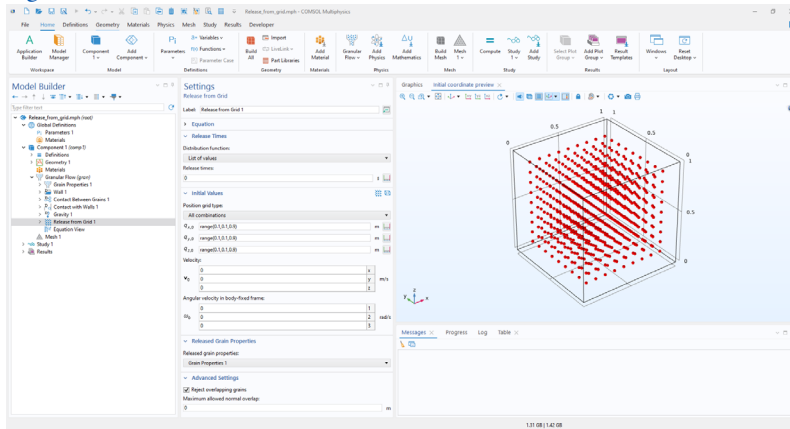
If **All combinations** is selected, a total of 16 grains will be released, including every possible combination of the initial  $x$ - and  $y$ -coordinates. If **Specified combinations** is selected, 4 grains will be released with initial positions (0,2), (1,4), (2,6), and (3,8).

The position for any grains with initial coordinates outside the geometry are set to not-a-number (NaN), so the grains do not appear when plotted during results processing.

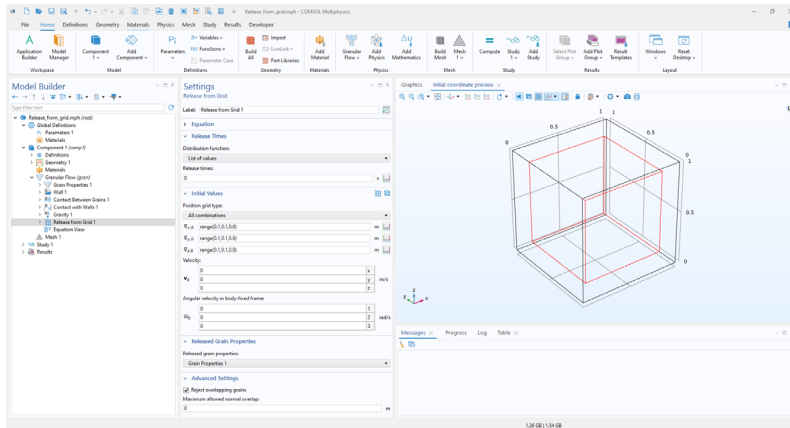
### *Previewing Grid Points*

In the **Initial Coordinates** section, click the **Preview Initial Coordinates**  and **Preview Initial Extents**  buttons to visualize the initial grain positions. Clicking **Preview Initial Coordinates** causes a point to appear in the **Graphics** window for every release position. Clicking **Preview Initial Extents** causes a bounding box to appear, indicating

the spatial extents of the released grains. Examples are shown in [Figure 3-1](#) and [Figure 3-2](#).



*Figure 3-1: Graphics window after clicking the Preview Initial Coordinates button.*



*Figure 3-2: Graphics window after clicking the Preview Initial Extents button.*

## RELEASED GRAIN PROPERTIES

Choose the desired set of grain properties described in one of the **Grain Properties** nodes from **Released grain properties** list. The default is the first **Grain Properties** node in the node tree.

## ADVANCED SETTINGS

By default, grains will be released at the initial positions only if they do not interact with any existing grains or with the walls. This can be controlled by the **Reject overlapping grains** checkbox which is selected by default. When enabled, enter a nonnegative value or expression for the **Maximum allowed normal overlap** (SI unit: m). The default is 0 m. This value is used to control the amount of acceptable overlap. When the checkbox is not selected, grains are always released at the specified coordinates irrespective of the overlap with other grains and wall elements.

If **User Defined** is selected from the **Seeds for random number generator** list in the physics interface **Advanced Settings** section, the **Seed for random number generator** field is available. Enter the seed value of the pseudorandom number generator (PRNG) used by this feature. The default value is 1. The PRNG is used to generate random numbers for sampling grain release times, positions, grain properties, and initial values of auxiliary dependent variables.

### *Release from Data File*

---

Use a **Release from Data File** node to release grains by specifying the initial position, velocity, and values of auxiliary dependent variables using data from a text file.

See [Release](#) for information about the following sections: **Release Times** and **Initial Value of Auxiliary Dependent Variables**.

See [Release from Grid](#) for information about the following sections: **Released Grain Properties** and **Advanced Settings**.

### *Formatting Guidelines for Files Containing Grain Data*

The imported data file should be a text file (\*.txt) arranged in a spreadsheet format; that is, each row corresponds to a distinct grain and should have the same number of columns as all other rows. Columns can be separated by spaces, tabs, or a combination of the two. Begin a line with the percent (“%”) character to include comments or empty lines in the data file.

For example, a data file containing the following text would insert grains at the positions (0.1, 0.2, 0.6) and (0.2, 0.4, 0.8) in a three-dimensional geometry:

```
% Initial grain positions
% qx0 qy0 qz0
0.1 0.2 0.6
0.2 0.4 0.8
```

## INITIAL VALUES

See the [Release](#) node for information about **Angular velocity in body-fixed frame** and **Temperature**.

### *Filename*

**Browse** your computer's file system to select a text file, then click **Import** to import the data. To remove the imported data, click **Discard**. Enter the **Index of first column containing position data**  $i$  to indicate which column represents the first coordinate of the grain position vectors. The default value 0 indicates the first column.

### *Velocity*

Select an option from the **Initial velocity** list: **From file**, or **User defined** (the default).

- For **From file**, enter the **Index of first column containing velocity data**  $i$ . The default is 3 in 3D and 2 in 2D. The columns are zero-indexed; that is, an index of 0 corresponds to the first column. Select the **Rotate velocity vectors** checkbox to rotate the velocity vectors using the specified **Euler angles (Z-X-Z)** in 3D or **Rotation angle** in 2D. The checkbox is cleared by default.
- For **User defined**, enter values or expressions for the **Initial grain velocity**  $\mathbf{v}_0$  (SI unit: m/s) based on space dimension. The defaults are 0 m/s.

## TRANSFORMATIONS

The distribution of loaded grain positions can be scaled, rotated, and translated before the grains are released.

To scale the distribution of release positions, enter a value or expression for the **Scale factor**  $R$  (dimensionless). The default is 1. This scale factor can be used to correct unit discrepancies between the data file and the model geometry. For example, if the geometry length unit is in meters but the data file lists coordinates in millimeters, enter a scale factor of 0.001.

To rotate the distribution, enter the **Euler angles (Z-X-Z)**  $\alpha$ ,  $\beta$ , and  $\gamma$  (in 3D) or the **Rotation angle**  $\alpha$  (in 2D). The default values are all 0.

In 3D,  $\alpha$  is the rotation angle about the space-fixed  $z$ -axis, then  $\beta$  is the rotation angle about the transformed  $x$ -axis (or  $x'$ -axis), and finally  $\gamma$  is the rotation angle about the transformed  $z$ -axis (or  $z''$ -axis). Positive values indicate counterclockwise rotations.

Enter values or expressions for the components of the **Displacement** vector  $\Delta\mathbf{q}$  (SI unit: m). The default is not to apply any translation.

If a translation and another type of transformation (scaling or rotation) are applied, then the translation is always applied after the other transformations.

### *Outlet*

---

Use the **Outlet** node to determine what happens to the grains when passing through the selected boundaries. The [Accumulator \(Boundary\)](#) subnode is available from the context menu (right-click the parent node) or from the **Attributes** menu on the **Physics** toolbar.

#### **OUTLET**

Select a **Wall condition**: **Disappear** or **Pass through** (the default).

- When **Disappear** is selected, the grains passing through the boundaries are removed from the modeling domain.
- By default, any common boundary between two adjacent domains are treated as rigid walls. Use the **Pass through** condition to allow the grains to pass from one domain to the other.



The Granular Flow interface also provides the [Bounding Box](#) feature to remove grains. For performance reasons, it is recommended to use the [Bounding Box](#) instead of the **Disappear** option when possible.

---

### *Force*

---

Use the **Force** node to exert user-defined external forces on grains to influence their motion. All forces defined in the model are added together to compute the total force on the grains.

#### **FORCE**

Enter values or expressions for the components of **Force  $\mathbf{F}$**  (SI unit: N) based on space dimension. The defaults are 0 N.

### *Torque*

---

Use the **Torque** node to exert user-defined torque on grains. All torque defined in the model are added together to compute the total torque on the grains.

## TORQUE

Enter values or expressions for the **Torque in body-fixed frame**  $\tau$  (SI unit: Nm) in 3D. The default values are 0 Nm.

In 2D, enter values or expressions for the **Grain Torque**  $\tau$  (SI unit: Nm). The default value is 0 Nm.

## *Auxiliary Dependent Variable*

---

Use the **Auxiliary Dependent Variable** node to add additional degrees of freedom for each grain. These can be used to solve for the path length, residence time, number of wall collisions, spin, and so forth.

## AUXILIARY DEPENDENT VARIABLE

Enter a **Field variable name**. The default is `rp` and can be changed to anything provided it does not conflict with the name of the variables for the position or momentum degrees of freedom. The name should not conflict with other auxiliary dependent variables.

Enter a **Source**  $R$ . The unit of the source depends on the settings in the **Units** section.

Under **Integrate** choose whether to integrate the equation you have defined **With respect to time** (the default) or **Along grain trajectory**. For example, to compute the residence time of a group of grains in a given system, set the **Source** to 1 and set **Integrate** to **With respect to time**. To compute the length of the grain trajectory, set the **Source** to 1 and set **Integrate** to **Along grain trajectories**.

## UNITS

Select a **Dependent variable quantity** from the list; the default is **Dimensionless [1]**. To enter a unit, select **None** from the list and in the **Unit** field enter a value (for example, K, m/s, or mol/m<sup>3</sup>).



[Auxiliary Dependent Variables Theory](#)

---

## *Periodic Condition*

---

Use the **Periodic Condition** node to apply periodic boundary conditions along one of the  $x$ ,  $y$ , or  $z$  directions. The **Periodic Condition** feature requires the selection of two sets of planar boundaries: source and destination.

Grains close to the source (destination) boundaries can interact with the periodic images of the grains near the destination (source) boundaries. Further, a grain crossing the source (destination) boundaries will automatically be placed at the corresponding locations on the destination (source) boundaries.

### **BOUNDARY SELECTION**

The selection of the boundaries is important and needs to satisfy the following criteria for the periodic boundary conditions to work correctly.

- The source and the destination boundaries need to be planar. Curved boundaries are not supported. Further, the normals of the selected boundaries must align with one of the  $x$ -,  $y$ -, or  $z$ -axes.
- If a **Periodic Condition** is applied along one axis, then all the boundaries in the geometry whose normals lie along that axis must be selected. Periodic boundary conditions cannot be selectively applied on a part of the geometry along a chosen direction.
- The source and destination boundaries must also be the exterior boundaries of the geometry along the chosen direction.



The software usually automatically identifies the boundaries as either source boundaries or destination boundaries, as indicated in the selection list. This works fine for cases like opposing parallel boundaries. In other cases, right-click **Periodic Condition** and select **Manual Destination Selection** to control the destination. By default it contains the selection that COMSOL Multiphysics identifies.

---

### **DESTINATION SELECTION**

This section is available for specifying the destination boundaries, if needed, when the **Manual Destination Selection** (right-click the parent node) option is selected in the context menu for the **Periodic Condition** node. You can only select destination boundaries from the union of all source and destination boundaries.

## *Grain Counter*

---

Use the **Grain Counter** feature to compute information about grains that are located in a set of selected domains. The counter can include all grains or may be restricted to grains corresponding to a specific release feature or a **Grain Properties** node. Computed variables are the number of grains transmitted, the number of grains transmitted at the final time, the transmission probability, and a logical expression which can be used to filter the rendered grains during results processing.

### **GRAIN COUNTER**

Select an option from the **Grain selection** list to specify whether the **Grain Counter** collects information based on **Release feature** (the default) or **Grain properties**.

If **Release feature** is selected, select an option from the **Release feature** list. If **All** (the default) is selected, the **Grain Counter** collects information about all grains in the selected domains, regardless of how they were released. Alternatively, select a grain release feature from the list, and then only the grains produced by that release feature are counted.

If **Grain properties** is selected from the **Grain selection** list, select an option from the **Released grain properties** list. If **All** (the default) is selected, the **Grain Counter** collects information about all grains in the selected domains, regardless of their grain properties. Alternatively, select a grain properties feature from the list, and then only the grains associated with the selected **Grain Properties** node are counted.

## *Bounding Box*

---

Use the **Bounding Box** feature to add a virtual rectangular box that bounds the modeling domains. Any grains that cross the bounds of this box are removed from the simulation. This feature can be used to discard the grains that move away from the region of interest since these grains may otherwise have an adverse effect on the performance of the simulation. Only one **Bounding Box** node can be added to the model.

### **SETTINGS**

Select an option from the **Specify bounding box** list: **From geometry** or **User defined** (default).

- If **From geometry** is selected, a grain is immediately removed if it reaches the surface of the bounding box of the geometry, defined as the smallest rectangle or rectangular prism with edges parallel to the axes of the global coordinate system that

encloses all of the geometric entities. This bounding box is extended by 5% in each direction so that the grains can still interact with boundaries that coincide with the perimeter of the bounding box.

- If **User defined** is selected, enter the values of the **x minimum**, **x maximum**, **y minimum**, and **y maximum**. In 3D components, also enter the values of **z minimum** and **z maximum**. A grain is immediately terminated if it reaches the surface of a box of user-specified dimensions.

### *Heat Source*

---

The **Heat source** node is available when the **Compute grain temperature** checkbox is selected in the physics interface node's **Additional Variables** section. Use this node to apply a user-defined source or sink term that affects the grain temperature.

#### **HEAT SOURCE**

Enter a **Heat source**  $Q$  (SI unit: W). The default value is 0.

### *Convective Heat Transfer*

---

The **Convective Heat Transfer** node is available when the **Compute grain temperature** checkbox is selected in the physics interface node's **Additional Variables** section. Use the **Convective Heat Transfer** node to model the heating or cooling of a grain due to convective heat exchange with the surrounding fluid, assuming surrounding fluid has no effect on the motion of grains.

The temperature within the grain is assumed to be uniform; that is, heat transfer by conduction within the grain takes place on a much shorter time scale than heat transfer by convection at the surface. This is equivalent to the assumption that the grain Biot number is much smaller than unity, and allows each grain's temperature to be stored as a single number instead of a temperature distribution.

#### **MODEL INPUT**

The model input for the **Temperature**  $T$  (SI unit: K) is always shown in the settings window for this feature, even if there are no material properties that depend on it.

#### **CONVECTIVE HEAT TRANSFER**

Choose an option from the **Heat source definition** list: **Specify heat transfer coefficient** or **Specify Nusselt number** (the default).

If **Specify heat transfer coefficient** is selected, enter a value or expression for the **Heat transfer coefficient**  $h$  (SI unit:  $\text{W}/(\text{m}^2\cdot\text{K})$ ). The default is  $10 \text{ W}/(\text{m}^2\cdot\text{K})$ .

If **Specify Nusselt number** is selected, then by default the **Thermal conductivity**  $k$  (SI unit:  $\text{W}/(\text{m}\cdot\text{K})$ ) is taken **From material**. Select **User defined** from the list to enter a value or expression. The default is  $0.025 \text{ W}/(\text{m}\cdot\text{K})$ . For this purpose, the conductivity is always assumed isotropic. Also enter a value or expression for the dimensionless **Grain Nusselt number**  $\text{Nu}$ . The default value, 2, is the theoretical limit for a sphere in a fluid with zero relative velocity.



Convective Heat Transfer in the theory section.

---

### *Accumulator (Boundary)*

---

The **Accumulator** subnode is available from the context menu (right-click the **Wall** node) or from the **Physics** toolbar, **Attributes** menu. Each **Accumulator** subnode defines a variable, called the accumulated variable, on each boundary element in the selection of the parent node. Whenever a grain hits a boundary element, the value of the accumulated variable in that element is incremented based on the value of the user-defined **Source** term  $R$  for the incident grain. The accumulated variable can be influenced by all grains or be restricted to grains corresponding to a specific release feature or a **Grain Properties** node.

#### **ACCUMULATOR SETTINGS**

Select an option from the **Accumulator type** list: **Density** (default) or **Count**.

- For **Density** the accumulated variable is divided by the surface area (in 3D) or length (in 2D) of the boundary element where it is defined.
- For **Count** the accumulated variable is the sum of the source terms of all grains that hit the boundary element and is unaffected by the boundary element size.

Enter the **Accumulated variable name**. The default is `rpb`. The accumulated variable is defined as `<scope>.<name>`, where `<scope>` includes the name of the physics interface node, parent boundary condition, and the **Accumulator** node; and `<name>` is the accumulated variable name.

For example, if the **Accumulator** subnode is added to a **Wall** node in an instance of the Granular Flow interface using the default variable name `rpb`, the accumulated variable name might be `gran.wp.bacc1.rpb`.

Enter a **Source  $R$** . The unit of the source term depends on the settings in the **Units** section. Whenever a grain comes in contact with a boundary element in the selection of the parent node, the accumulated variable in that element is incremented by the source term. If the **Accumulator type** is set to **Density**, the source term is divided by the area of the boundary element (in 3D) or the length of the boundary element (in 2D).

### GRAIN SELECTION

Select an option from the **Grain selection** list to specify whether the accumulated variable collects information from the grains based on a **Release feature** (the default) or a **Grain properties** node.

If **Release feature** is selected, select an option from the **Release feature** list. If **All** (the default) is selected, the accumulated variable collects information about all grains in the selected domains or boundaries, regardless of how they were released. Alternatively, select a grain release feature from the list, and then only the grains released by that release feature are included.

If **Grain properties** is selected from the **Grain selection** list, select an option from the **Released grain properties** list. If **All** (the default) is selected, the accumulated variable collects information about all grains in the selected domains or boundaries, regardless of their grain properties. Alternatively, select a grain properties feature from the list, and then only the grains associated with the selected **Grain Properties** node are included.

### UNITS

Select a **Dependent variable quantity** from the list; the default is **Dimensionless [1]**. To enter a unit, select **None** from the list and in the **Unit** field enter a value, for example, K, m/s, or mol/m<sup>3</sup>.

### SMOOTHING

The accumulated variables are computed using discontinuous shape functions. Select the **Compute smoothed accumulated variable** checkbox to compute a smoothed accumulated variable by computing the average value of the variable within a sphere of a user-defined radius. Then enter a **Smoothing radius  $r$**  (SI unit: m). The default is 0.1 m.



## *Force Accumulator*

---

The **Force Accumulator** subnode is available from the context menu (right-click the **Wall** node) or from the **Physics** toolbar, **Attributes** menu. Each **Force Accumulator** subnode is a special type of boundary accumulator that accumulates normal contact force on the wall element during the contact between the grain and the wall.

### **ACCUMULATOR SETTINGS**

Enter the **Accumulated variable name**. The default is `wf`. The accumulated variable is defined as `<scope>.<name>`, where `<scope>` includes the name of the physics interface node, parent boundary condition, and the **Force Accumulator** node; and `<name>` is the accumulated variable name.

For example, if the **Force Accumulator** subnode is added to a **Wall** node in an instance of the Granular Flow interface using the default variable name `wf`, the accumulated variable name might be `gran.wp.facc1.wf`.

### **GRAIN SELECTION**

See [Accumulator \(Boundary\)](#) for settings related to **Grain Selection**.



[Force Accumulator Theory](#)

---

## *Accumulator (Domain)*

---

Use the **Accumulator** node to define additional degrees of freedom on a domain. Each **Accumulator** defines a variable, called the accumulated variable, on each domain element in the set of selected domains. The values of the accumulated variables are determined by the properties of grains in each domain element. The accumulated variable may be influenced by all grains or may be restricted to grains corresponding to a specific release feature or a **Grain Properties** node.

### **ACCUMULATOR SETTINGS**

Select an option from the **Accumulator type** list: **Density** (default) or **Count**.

- For **Density**, the accumulated variable is divided by the volume of the mesh element where it is defined.
- For **Count**, the accumulated variable is unaffected by the element size.

Select an option from the **Accumulate over** list: **Elements** (default) or **Elements and time**.

- For **Elements**, the value of the accumulated variable in an element is the sum of the source terms of all grains in that element. If the **Accumulator type** is set to **Density**, this sum is divided by the mesh element volume. At a later time, a grain has no effect on the value of the accumulated variable in an element it passed through previously.
- For **Elements and time**, the time derivative of the accumulated variable in an element is the sum of the source terms of all grains in that element. If the **Accumulator type** is set to **Density**, this sum is divided by the mesh element volume. As each grain moves through a series of mesh elements, it leaves behind a contribution to the accumulated variable that remains even after the grain has moved on.

Enter the **Accumulated variable name**. The default name is `rpd`. The accumulated variable is defined as `<name>.<varname>`, where `<name>` is the physics interface name and `<varname>` is the accumulated variable name. For example, in an instance of the Granular Flow interface with default name `gran` and default accumulated variable name `rpd`, the variable would be named `gran.rpd`.

Enter a **Source  $R$** . The unit of the source depends on the settings in the **Units** section. The source term is used to calculate the accumulated variable in a manner specified by the **Accumulate over** and **Accumulator type** settings.

If **Elements and time** is selected from the **Accumulate over** list, select an option from the **Source interpolation** list: **Constant**, **Linear** (the default), **Quadratic**, or **Exponential**. The **Source interpolation** determines what functional form the **Source** is assumed to follow during each time step taken by the solver. This information is used to compute the accumulated variable in mesh elements that the grains pass through during each time step.

## GRAIN SELECTION

See [Accumulator \(Boundary\)](#) for settings related to **Grain Selection**.

## UNITS

Select a **Dependent variable quantity** from the list; the default is **Dimensionless [1]**. To enter a unit, select **None** from the list and in the **Unit** field enter a value, for example, `K`, `m/s`, or `mol/m^3`.



# Theory for the Granular Flow Interface

The Granular Flow Interface theory is described in this section:

- Kinematics of Grains
- Contact Force: Linear Elastic Model
- Contact Force: Hertz–MD (Mindlin and Deresiewicz) Model
- Contact Force: Hertz–MD with Adhesion Model
- Contact Force: van der Waals Force
- Contact Force: Coulomb’s Criterion
- Rotational Resistance Theory
- Computing Grain Temperature
- Contact Search Theory
- Initial Conditions
- Boundary Conditions
- Auxiliary Dependent Variables Theory
- Accumulator Theory: Domains
- Accumulator Theory: Boundaries
- Force Accumulator Theory
- Time Step Size
- References

## *Kinematics of Grains*

---

A grain has two types of motion: translational and rotational motion. These motions of individual grains are determined by the equations of the motion given by

$$m_i \frac{d\mathbf{v}_i}{dt} = \sum_j (\mathbf{F}_{n,ij} + \mathbf{F}_{t,ij}) + \mathbf{F}_{\text{ext},i} \quad (3-1)$$

$$I_i \frac{d\omega_i}{dt} = \sum_j (\mathbf{R}_i \times \mathbf{F}_{t,ij} + \mathbf{M}_{\text{rot},ij}) \quad (3-2)$$

where

- $m$ ,  $I$ ,  $\mathbf{v}$ , and  $\omega$  are the mass, moment of inertia, translational velocity, and rotational velocity of the grain, respectively.
- $\mathbf{F}_n$  and  $\mathbf{F}_t$  are the normal and the tangential forces during contact between grains  $i$  and  $j$ ; for example, [Contact Force: Hertz–MD \(Mindlin and Deresiewicz\) Model](#).
- $\mathbf{R}_i = R_i \mathbf{n}_{ij}$  is the vector between the center of the grain and the contact point where the force  $\mathbf{F}_t$  is applied, with  $R_i$  the radius and  $\mathbf{n}_{ij}$  the unit normal vector along the line joining the center and point of contact.
- $\mathbf{F}_{\text{ext}}$  are all other external forces applied to the grain such as gravitational force.
- $\mathbf{M}_{\text{rot}}$  is the torque due to rotational friction and resists the rotation of grain; see [Rotational Resistance Theory](#) for more details.

For the orientation of grain, a quaternion  $Q$  which is a vector in 4D space is used ([Ref. 1](#)). Assuming  $\phi$ ,  $\theta$ , and  $\psi$  to be the Euler angles defined in the body-fixed frame, then the components of quaternions are related to Euler's angles as

$$Q = \begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix} = \begin{bmatrix} \cos \frac{\theta}{2} \cos \frac{\phi + \psi}{2} \\ \sin \frac{\theta}{2} \cos \frac{\phi - \psi}{2} \\ \sin \frac{\theta}{2} \sin \frac{\phi - \psi}{2} \\ \cos \frac{\theta}{2} \sin \frac{\phi + \psi}{2} \end{bmatrix} \quad (3-3)$$

The angular velocity in the body-fixed frame can be then evaluated as

$$\dot{Q} = \begin{bmatrix} \dot{q}_0 \\ \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{bmatrix} = \frac{1}{2} \begin{bmatrix} -q_1 \omega_1 - q_2 \omega_2 - q_3 \omega_3 \\ q_0 \omega_1 + q_2 \omega_3 - q_3 \omega_2 \\ q_0 \omega_2 - q_1 \omega_3 + q_3 \omega_1 \\ q_0 \omega_3 + q_1 \omega_2 - q_2 \omega_1 \end{bmatrix} \quad (3-4)$$

Newton's second law of motion ([Equation 3-1](#)) is expressed as a set of coupled first-order ordinary differential equations with  $\mathbf{F}$  as the total force:

$$m \dot{\mathbf{v}} = \mathbf{F}$$

$$\mathbf{v} = \dot{\mathbf{q}}$$

Similarly, the rotational motion is given as with  $\Gamma$  as the total torque

$$I\dot{\omega} = \Gamma$$

and  $\omega$  is calculated using Equation 3-4 in 3D, while in 2D it is calculated as the derivative of the orientation  $\theta$  of grain:

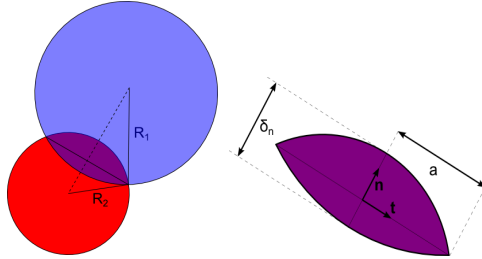
$$\omega = \dot{\theta}$$

### *Contact Force: Linear Elastic Model*

Select the **Linear elastic** from the **Contact Force Model** list in the physics interface **Force** section to compute the contact forces using the linear elastic model. The two components of forces are discussed individually (Ref. 1).

#### **NORMAL FORCE**

The following diagram (left) shows two grains in contact with radii  $R_1$  and  $R_2$ . As this is a soft-sphere model, there is a finite overlap between the grains in contact, although for illustrative purposes this overlap region has been greatly exaggerated.



Taking a closer look at the overlap region (right figure), define unit vectors in the tangential direction  $\mathbf{t}$  and normal direction  $\mathbf{n}$ . In 3D, there would be two orthogonal tangential directions  $\mathbf{t}_1$  and  $\mathbf{t}_2$ . Let the normal displacement  $\delta_n$  (SI unit: m) be the thickness of this overlap region. For intersecting spheres, the radius of the contact area is denoted  $a$  (SI unit: m). For two grains in contact with positions  $\mathbf{q}_i$  and  $\mathbf{q}_j$  (SI unit: m), the normal direction is

$$\mathbf{n}_{ij} = \frac{\mathbf{q}_j - \mathbf{q}_i}{|\mathbf{q}_j - \mathbf{q}_i|} \quad (3-5)$$

and the normal overlap is

$$\delta_{n,ij} = R_j + R_i - |\mathbf{q}_j - \mathbf{q}_i| \quad (3-6)$$

When the grains are in contact with each other, the normal component of the contact force is given by

$$\mathbf{F}_{n,ij} = -(k_{n,ij}\delta_{n,ij} + c_{n,ij}\mathbf{v}_{r,ij} \cdot \mathbf{n}_{ij})\mathbf{n}_{ij} \quad (3-7)$$

where  $k_n$  is the normal elastic stiffness coefficient and  $c_n$  is the normal damping coefficient. In this model, these two coefficients are known and constant.  $N_c$  is the number of neighboring grains, and  $\mathbf{v}_r$  is the relative velocity between colliding grains at the contact point and is given by

$$\mathbf{v}_{r,ij} = \mathbf{v}_i - \mathbf{v}_j + (R_i\omega_i + R_j\omega_j) \times \mathbf{n}_{ij}$$

where  $\omega$  is the rotational velocity of the grain. The normal damping coefficient  $c_n$  is calculated as

$$c_{n,ij} = -\log e_n \sqrt{\frac{4m_{\text{eff}}k_{n,ij}}{\pi^2 + (\log e_n)^2}}$$

where the effective mass is given by

$$m_{\text{eff}} = \left(\frac{1}{m_i} + \frac{1}{m_j}\right)^{-1}$$

and  $e_n$  is the coefficient of restitution in normal direction.

### TANGENTIAL FORCE

The tangential component of the contact force is given by

$$\mathbf{F}_{t,ij} = -(k_t\delta_{t,ij} + c_t\mathbf{v}_{r,ij} \cdot \mathbf{t}_{ij})\mathbf{t}_{ij} \quad (3-8)$$

where  $k_t$  is the tangential elastic stiffness coefficient and  $c_t$  is the tangential damping coefficient. Similar to normal force, these two are known and constant. The tangential damping constant is calculated as

$$c_{t,ij} = -\log e_t \sqrt{\frac{4m_{\text{eff}}k_{t,ij}}{\pi^2 + (\log e_t)^2}}$$

$e_t$  is the coefficient of restitution in tangential direction.

The tangential direction is given by

$$\mathbf{t}_{ij} = \frac{\mathbf{v}_{t,ij}}{|\mathbf{v}_{t,ij}|}$$

where  $\mathbf{v}_t$  is the tangential component of  $\mathbf{v}_r$ , which is obtained by subtracting normal component  $\mathbf{v}_n$  from  $\mathbf{v}_r$ :

$$\mathbf{v}_t = \mathbf{v}_r - \mathbf{v}_n$$

$$\mathbf{v}_n = (\mathbf{v}_r \cdot \mathbf{n})\mathbf{n}$$

Thus, Equation 3-7 and Equation 3-8 can be rewritten as

$$\mathbf{F}_{n,ij} = - \sum_{j=1}^{N_c} (k_n \delta_{n,ij} + c_n \mathbf{v}_{n,ij}) \quad (3-9)$$

$$\mathbf{F}_{t,ij} = - \sum_{j=1}^{N_c} (k_t \delta_{t,ij} + c_t \mathbf{v}_{t,ij}) \quad (3-10)$$

The calculation of the tangential displacement,  $\delta_t$ , is dependent on time history of the physical contact between two grains. When a new physical contact happens at time  $t_0$  between two grains,  $\delta_t$  is zero and is calculated as

$$\delta'_{t,ij} = \int_{t_0}^t \mathbf{v}_{t,ij} dt$$

$$\delta_{t,ij} = \delta'_{t,ij} - (\delta'_{t,ij} \cdot \mathbf{n}_{ij})\mathbf{n}_{ij}$$

The two steps in the calculation ensure that  $\delta_t$  is in the contact plane. At the end of contact  $\delta_t$  is set to zero.

### CONTACT FORCE BETWEEN GRAIN AND WALL

The contact force between grain and wall is same as the contact force between grains. The expressions that are different are discussed in this section. The normal overlap is given by

$$\delta_{n,iw} = R_i - |\mathbf{q}_w - \mathbf{q}_i| \quad (3-11)$$

where  $\mathbf{q}_w$  is the contact point between grain and wall.

The normal direction becomes

$$\mathbf{n}_{iw} = \frac{\mathbf{q}_w - \mathbf{q}_i}{|\mathbf{q}_w - \mathbf{q}_i|}$$

Similarly, the relative velocity for grain becomes

$$\mathbf{v}_{r,iw} = \mathbf{v}_i + R_i \omega_i \times \mathbf{n}_{iw}$$

and the effective mass becomes

$$m_{\text{eff}} = m_i$$

Substituting all of the above equations in [Equation 3-9](#) and [Equation 3-10](#), the normal and tangential components of the contact force between grain and wall are calculated.

### *Contact Force: Hertz–MD (Mindlin and Deresiewicz) Model*

---

Select the **Hertz–MD** from the [Contact Force Model](#) list in the physics interface **Force** section to compute the contact forces using the Hertz–MD model. The equations for the forces in [Contact Force: Linear Elastic Model](#) used the linear equations where the stiffness coefficients,  $k_n$  and  $k_t$ , and damping coefficients,  $c_n$  and  $c_t$  are known and constant. For the case where these coefficients are not known, a nonlinear method is used where the coefficients depend on the material properties and normal overlap between grains  $\delta_n$  ([Ref. 1](#)).

#### **NORMAL FORCE**

The equation of normal force remains same as [Equation 3-9](#); however, the spring and damping coefficients are not constant and depend on the material properties of grains  $i$  and  $j$  in contact:

$$\begin{aligned} k_{n,ij} &= \frac{4}{3} E_{\text{eq}} \sqrt{R_{ij} \delta_{n,ij}} \\ c_{n,ij} &= -\log e_n \sqrt{\frac{5m_{\text{eff}} k_{n,ij}}{\pi^2 + (\log e_n)^2}} \\ E_{\text{eq}} &= \left( \frac{1 - \nu_i^2}{E_i} + \frac{1 - \nu_j^2}{E_j} \right)^{-1} \end{aligned} \quad (3-12)$$

$$R_{\text{eq}} = \left( \frac{1}{R_i} + \frac{1}{R_j} \right)^{-1} \quad (3-13)$$

$$m_{\text{eff}} = \left( \frac{1}{m_i} + \frac{1}{m_j} \right)^{-1}$$

where

- $k_n$  is the normal spring stiffness
- $c_n$  is the normal damping
- $e_n$  is the coefficient of restitution in normal direction
- $E$  is the Young's modulus
- $\nu$  is the Poisson's ratio
- $m$  and  $R$  are the mass and radius of grain respectively
- $E_{\text{eq}}$  is the equivalent Young's modulus for grains in contact
- $R_{\text{eq}}$  is the equivalent radius for grains in contact
- $m_{\text{eff}}$  is the effective mass for grains in contact

#### TANGENTIAL FORCE

The equation of normal force remains same as [Equation 3-10](#); however, the spring and damping coefficients are defined as

$$k_{t,ij} = 8G_{\text{eq}}\sqrt{R_{\text{eq}}\delta_{n,ij}}$$

$$c_{t,ij} = -\mathbf{log}e_t \sqrt{\frac{10}{3} \frac{m_{\text{eff}}k_{t,ij}}{\pi^2 + (\mathbf{log}e_t)^2}}$$

$$G_{\text{eq}} = \left( \frac{2-\nu_i^2}{G_i} + \frac{2-\nu_j^2}{G_j} \right)^{-1} \quad (3-14)$$

where

- $k_t$  is the tangential spring stiffness
- $c_t$  is the tangential damping
- $e_t$  is the coefficient of restitution in tangential direction
- $G$  is the shear modulus
- $G_{\text{eq}}$  is the equivalent shear modulus for grains in contact

### CONTACT FORCE BETWEEN GRAIN AND WALL

The contact force between grain and wall is the same as the contact force between grains. The expressions that are different are discussed in this section. For the contact between grain  $i$  and wall  $w$ ,

$$E_{\text{eq}} = \left( \frac{1 - \nu_i^2}{E_i} + \frac{1 - \nu_w^2}{E_w} \right)^{-1} \quad (3-15)$$

$$G_{\text{eq}} = \left( \frac{2 - \nu_i^2}{G_i} + \frac{2 - \nu_w^2}{G_w} \right)^{-1} \quad (3-16)$$

$$R_{\text{eq}} = R_i \quad (3-17)$$

$$m_{\text{eff}} = m_i$$

#### *Contact Force: Hertz–MD with Adhesion Model*

Select the **Hertz–MD with Adhesion** from the **Contact Force Model** list in the physics interface **Force** section to compute the contact forces using the Hertz–MD with adhesion model. A widely used adhesive contact force model used in DEM is the Johnson–Kendall–Roberts (JKR) contact model, which is an extension of the Hertz–MD contact force model and allows for tensile force to develop due to surface adhesion,  $\gamma$ , which is calculated here as (Ref. 2)

$$\gamma = \frac{\gamma_i + \gamma_j - 2\gamma_{ij}}{2}$$

where  $i$  and  $j$  are indices for the grains in contact and  $\gamma_{ij}$  is the interface energy density.

### NORMAL FORCE

The normal contact force is given by (Ref. 3)

$$\mathbf{F}_{\text{n},ij} = - \left( \frac{4E_{\text{eq}}a_{ij}^3}{R_{\text{eq}}} - \sqrt{16\pi\gamma E_{\text{eq}}a_{ij}^3 + c_{\text{n},ij}\mathbf{v}_{\text{r},ij} \cdot \mathbf{n}_{ij}} \right) \mathbf{n}_{ij} \quad (3-18)$$

where  $a$  is the radius of the circular contact patch area. For the definition of other parameters used, see **Contact Force: Hertz–MD (Mindlin and Deresiewicz) Model** and **Contact Force: Linear Elastic Model**.

An analytical solution to calculate  $a$  for given  $\delta_n$  is given in [Ref. 4](#). The contact force between grains (or between grain and wall) starts when there is a physical contact and the contact force remains active as the gap between the neighboring grains is less than or equal to the cutoff distance,  $\delta_c$ , which is given by ([Ref. 3](#)),

$$\delta_c = \frac{1}{2} \frac{1}{6^{1/3}} \frac{a_0^2}{R_{eq}}$$

where  $a_0$  is the equilibrium contact patch radius when  $|\mathbf{F}_n|=0$  and no other external forces are acting and is given by

$$a_0 = \left( \frac{9\pi\gamma R_{eq}^2}{E_{eq}} \right)^{1/3}$$

### TANGENTIAL FORCE

The tangential force is exactly the same as that of [Contact Force: Hertz–MD \(Mindlin and Deresiewicz\) Model](#) except the definition of  $k_t$ , which is given by

$$k_{t,ij} = 8G_{eq}a_{ij}$$

### *Contact Force: van der Waals Force*

---

Select the [Compute van der Waals Force](#) checkbox in the physics interface's **Force** section to compute the van der Waals force between grains and between grains and walls. The **Compute van der Waals force** checkbox is only available when **Hertz–MD** or **Hertz–MD with adhesion** is selected in the **Contact force model** list in the physics interface **Force** section. Generally when the grain sizes are less than 10 microns, contribution of van der Waals force becomes nonnegligible. In such cases, the van der Waals force is added to the normal component of contact force and is given by ([Ref. 4](#))

$$F_{vw} = \begin{cases} 4\pi\gamma R_{eq} & \text{if } \delta_n > 0 \\ \frac{4\pi\gamma R_{eq} D_{min}^2}{(\delta_n - D_{min})^2} & \text{if } -D_{max} \leq \delta_n \leq 0 \\ 0 & \text{if } \delta_n < -D_{max} \end{cases}$$

where

- $\gamma$  is surface energy per unit area
- $D_{\max}$  is the cutoff distance for the van der Waals force
- $D_{\min}$  is a parameter to avoid the singularity
- $\delta_n$  is normal overlapping distance and is defined using Equation 3-6 and Equation 3-11 between grains and between grain and wall, respectively
- $R_{\text{eq}}$  is the equivalent radius between grains in contact (Equation 3-13) or between grain and wall (Equation 3-17)

The van der Waals force is calculated as an addition to the normal force, Equation 3-7, as follows:

$$\mathbf{F}_{n,ij} = -(k_{n,ij}\delta_{n,ij} - F_{\text{vw}} + c_{n,ij}\mathbf{v}_{r,ij} \cdot \mathbf{n}_{ij})\mathbf{n}_{ij}$$

The effect on the tangential force component is not considered in the current formulation of the van der Waals force.

### *Contact Force: Coulomb's Criterion*

---

Coulomb's criterion is used to prevent the unphysical sliding between grains or between grain and wall. The gross-sliding occurs when Coulomb's criterion is violated or achieved; the criterion is (Ref. 1)

$$|\mathbf{F}_t| \geq \mu_s |\mathbf{F}_n| \quad (3-19)$$

where  $\mu_s$  is the coefficient of sliding friction. Hence, to not violate Coulomb's criterion, the following form of the tangential force is used:

$$\mathbf{F}_t = -\min(k_t \cdot |\delta_t|, \mu_s |\mathbf{F}_n|) \frac{\delta_t}{|\delta_t|} \quad (3-20)$$

When Coulomb's criterion is satisfied, the tangential displacement is limited as

$$\delta_t \leftarrow \frac{\mu_s |\mathbf{F}_n|}{k_t} \frac{\delta_t}{|\delta_t|} \quad (3-21)$$

Equation 3-19 to Equation 3-21 are true when the contact force model is either linear elastic or Hertz–MD. However, for the [Contact Force: Hertz–MD with Adhesion Model](#) these equations need to be slightly modified to account for the cohesive effects.

A separation force,  $F_c$  is needed to separate two grains in equilibrium with no external force present and is given by

$$F_c = 3\pi\gamma R_{eq} \quad (3-22)$$

So, now Coulomb's criterion becomes

$$|\mathbf{F}_t| \geq \mu_s(|\mathbf{F}_n| + 2F_c)$$

Accordingly, the tangential force needs to be updated by the relation

$$\mathbf{F}_t = -\min(k_t \cdot |\delta_t|, \mu_s(|\mathbf{F}_n| + 2F_c)) \frac{\delta_t}{|\delta_t|}$$

And the tangential displacement is limited as

$$\delta_t \leftarrow \frac{\mu_s(|\mathbf{F}_n| + 2F_c)}{k_t} \frac{\delta_t}{|\delta_t|}$$

### *Rotational Resistance Theory*

---

The rotational resistance torque arises from the unevenness of the contact pressure distribution in the contact area between rotating grains, or between a single rotating grain and a wall.



For example, for a stationary spherical grain sitting on a tabletop, the pressure distribution (normal contact force per unit area) in the soft-sphere model might look like the figure shown on the left. The pressure distribution is symmetric. However, when the grain is rolling as shown on the right, the pressure distribution becomes asymmetric, creating a net torque that opposes the rotation.

The rolling resistance is  $\mathbf{M}_{rot}$  in [Equation 3-2](#) and is implemented based on [Ref. 5](#).

#### **CONSTANT TORQUE MODEL**

Select the **Constant torque model** from the **Rotational resistance model** list in the physics interface's [Rotational Resistance](#) section to apply the rotational resistance torque using

the constant torque model. The rotational resistance torque in 2D is simply rolling resistance torque and is defined as

$$\mathbf{M}_{\text{rot}} = -\min\left(\mu_r \tau, I \frac{|\omega|}{\Delta t}\right) \hat{\omega}$$

where

- $\mu_r$  is the coefficient of rolling friction
- $\tau = R_{\text{eq}} |\mathbf{F}_n|$
- $\hat{\omega} = (\omega_i - \omega_j) / |\omega_i - \omega_j|$  is the unit vector of the relative angular velocity of the grains
- $R_{\text{eq}} = R_i R_j / (R_i + R_j)$  for contact between the grains,  $R_{\text{eq}} = R_i$  for contact with the walls
- $\Delta t$  is the time step size

In 3D, the rotational resistance torque has two components rolling resistance torque and twisting resistance torque and is defined as

$$\mathbf{M}_{\text{rot}} = -\min\left(\mu_r \tau, I \frac{|\omega_r|}{\Delta t}\right) \hat{\omega}_r - \min\left(\mu_{\text{tw}} \tau, I \frac{|\omega_{\text{tw}}|}{\Delta t}\right) \hat{\omega}_{\text{tw}}$$

where

$$\begin{aligned} \omega_{\text{tw}} &= (\omega_{ij} \cdot \mathbf{n}) \mathbf{n} \\ \omega_r &= \omega_{ij} - \omega_{\text{tw}} \\ \omega_{ij} &= \omega_i - \omega_j \end{aligned} \tag{3-23}$$

and  $\mathbf{n}$  is given by [Equation 3-5](#).

For [Contact Force: Hertz-MD with Adhesion Model](#),

$$\tau = R_{\text{eq}} (|\mathbf{F}_n| + 2F_c)$$

where  $F_c$  is given by [Equation 3-22](#).

#### VARYING TORQUE MODEL

Select the **Varying torque model** from the **Rotational resistance model** list in the physics interface's [Rotational Resistance](#) section to apply the rotational resistance torque using the varying torque model. The rotational resistance moment in 2D is just the rolling resistance torque and is defined in an incremental way as

$$\mathbf{M}_{\text{rot}, t + \Delta t} = \mathbf{M}_{\text{r}, t} + \Delta \mathbf{M}_{\text{r}}$$

The incremental torque is calculated from the incremental relative rotation between two grains and the rolling stiffness  $k_r$

$$\Delta \mathbf{M}_r = -k_{\text{rot}}(\omega_i - \omega_j)\Delta t$$

Here,  $k_r$  relates to tangential spring coefficient  $k_t$  and  $R_{\text{eq}}$  as

$$k_{\text{rot}} = k_t R_{\text{eq}}^2 \quad (3-24)$$

This choice sets the nominal rotational natural frequency due to rolling stiffness equal to the nominal rotational natural frequency due to the tangential or shear stiffness, leading to a well-behaved and well-damped rolling resistance mechanism without the need for any additional damping parameters.

The magnitude of the updated rolling resistance cannot be greater than the torque given by constant torque model

$$\mathbf{M}_r = \begin{cases} \mathbf{M}_r & \text{if } |\mathbf{M}_r| < M_{\text{rm}} \\ M_{\text{rm}} \frac{\mathbf{M}_r}{|\mathbf{M}_r|} & \text{otherwise} \end{cases} \quad (3-25)$$

where

$$\begin{aligned} M_{\text{rm}} &= \mu_r \tau \\ \tau &= R_{\text{eq}} |\mathbf{F}_n| \end{aligned}$$

In 3D, the rotational resistance can be decomposed into two component, rolling resistance and twisting resistance

$$\mathbf{M}_{\text{rot}, t + \Delta t} = \mathbf{M}_{r, t} + \Delta \mathbf{M}_r + \mathbf{M}_{\text{tw}, t} + \Delta \mathbf{M}_{\text{tw}}$$

$$\Delta \mathbf{M}_r = -k_{\text{rot}} \omega_r \Delta t$$

$$\Delta \mathbf{M}_{\text{tw}} = -k_{\text{rot}} \omega_{\text{tw}} \Delta t$$

$$\mathbf{M}_{\text{tw}} = \begin{cases} \mathbf{M}_{\text{tw}} & \text{if } |\mathbf{M}_{\text{tw}}| < M_{\text{twm}} \\ M_{\text{twm}} \frac{\mathbf{M}_{\text{tw}}}{|\mathbf{M}_{\text{tw}}|} & \text{otherwise} \end{cases}$$

$$M_{\text{twm}} = \mu_{\text{tw}} \tau$$

where  $k_{\text{rot}}$  is given by Equation 3-24,  $\omega_r$ ,  $\omega_{\text{tw}}$  are given by Equation 3-23, and  $\mathbf{M}_r$  is given by Equation 3-25.

For **Contact Force: Hertz–MD with Adhesion Model**,

$$\tau = R_{\text{eq}}(|\mathbf{F}_n| + 2F_c)$$

where  $F_c$  is given by Equation 3-22.

### *Computing Grain Temperature*

---

Select the **Compute Grain Temperature** checkbox in the physics interface’s **Grain Temperature** section to define an auxiliary dependent variable for the grain temperature. The grain temperature  $T_g$  (SI unit: K) is then computed along each grain trajectory by integrating the first-order equation

$$m_g C_{p,g} \frac{dT_g}{dt} = Q_t \quad (3-26)$$

where

- $m_g$  (SI unit: kg) is the grain mass,
- $C_{p,g}$  (SI unit: J/(kg·K)) is the grain specific heat capacity,
- $t$  (SI unit: s) is the time, and
- $Q_t$  (SI unit: W) is the sum of all heat sources and sinks affecting the grain.

#### **VALIDITY OF THE GRAIN TEMPERATURE CALCULATION**

The grain temperature is treated as a single value for each grain, not as a temperature distribution throughout the grain’s volume. Therefore, the temperature computation is only valid when the temperature throughout the grain can be considered uniform; that is, the heat transfer resistance within the grain is negligibly small compared to the heat transfer resistance at the surface of the grain. This is typically true for small grains with high thermal conductivity.

The Biot number  $\text{Bi}$  (dimensionless) can be used to determine whether the grain temperature can be treated as a uniform value. The Biot number is defined as

$$\text{Bi} = \frac{hL_C}{k_g}$$

where  $L_C$  (SI unit: m) is a characteristic length, typically the ratio of grain volume to grain surface area, and  $k_g$  (SI unit: W/(m·K)) is the grain thermal conductivity. If the

Biot number is very small, much less than unity, then the conductive heat transfer within the grain takes place on a much shorter time scale than convective heat transfer at the surface of the grain, so the grain temperature can be treated as a uniform value.

### CONTACT HEAT TRANSFER

Use the **Compute conductive heat transfer** checkbox in the physics interface [Additional Variables](#) section to apply the heat flux across the boundary between grains in contact and between grains and walls. **Compute conductive heat transfer** checkbox is only available when

- the **Hertz–MD** or the **Hertz–MD with adhesion** is selected from the **Contact force model** list in the physics interface **Force** section and
- the **Compute grain temperature** checkbox is selected in the physics interface [Additional Variables](#) section.

This heat flux contribution is added to total heat source  $Q_t$  in [Equation 3-26](#) and is defined as ([Ref. 6](#))

$$Q_{ij} = -4r_c \frac{k_{g,i} k_{g,j}}{k_{g,i} + k_{g,j}} (T_{g,j} - T_{g,i}) \quad (3-27)$$

where the contact radius is calculated as

$$r_c = C_r \left( \frac{3|\mathbf{F}_n|R_{eq}}{4E_{eq}} \right)^{\frac{1}{3}} \quad (3-28)$$

- $\mathbf{F}_n$  is the normal component of the contact force; see [Contact Force: Hertz–MD \(Mindlin and Deresiewicz\) Model](#), [Contact Force: Hertz–MD with Adhesion Model](#), or [Contact Force: van der Waals Force](#) for more details.
- $R_{eq}$  is calculated using [Equation 3-13](#).
- $E_{eq}$  is given by [Equation 3-12](#).
- $C_r$  is the temperature correction factor for contact radius.

The temperature correction factor,  $C_r$  equals to 1 when no correction is needed. It is a common practice in granular flow to use smaller value of Young’s modulus,  $\bar{E}$ , compared to the actual value,  $E_0$ , to simulate the bulk flow behavior. The main advantage is to be able to use larger time step without compromising the accuracy in bulk behavior of a granular system. However, taking smaller value of  $\bar{E}$  might result in overestimation of heat flux as the contact patch between grains or between grain and

surface increases. To fix this, usually a correction factor with following expression is used:

$$C_r = \left( \frac{E_{eq}}{E_{eq,0}} \right)^{\frac{1}{3}}$$

where  $E_{eq,0}$  is the equivalent Young's modulus calculated using real values of the Young's modulus of grains and walls.

Conductive heat transfer between grain and wall is calculated assuming the wall to have infinite radius and conductivity. Consequently, Equation 3-27 and Equation 3-28 become

$$Q_{iw} = - \sum_{j=1}^{N_w} 4r_c k_{g,i} (T_{w,j} - T_{g,i}) \quad (3-29)$$

$$r_c = C_r \left( \frac{3|F_n|R}{4E_{eq}} \right)^{\frac{1}{3}} \quad (3-30)$$

where  $E_{eq}$  is given by Equation 3-15.

### CONVECTIVE HEAT TRANSFER

Use the [Convective Heat Transfer](#) feature to apply convective heat transfer at the surface of the grains. This feature adds the following contribution to the total heat source  $Q_t$  in Equation 3-26:

$$Q = hA_g(T - T_g)$$

where

- $h$  (SI unit:  $W/(m^2 \cdot K)$ ) is the heat transfer coefficient,
- $A_g$  (SI unit:  $m^2$ ) is the grain surface area, and
- $T$  (SI unit:  $K$ ) is the temperature of the surrounding fluid at the grain's position.

Strictly speaking,  $T$  is the temperature that the surrounding fluid would have at the grain's position, if the grain were not there; the fluid very close to the surface of a warmer or cooler grain will show a temperature gradient. Assuming that the fluid temperature stays relatively constant over length scales comparable to the grain diameter, we can think of  $T$  as the ambient or free-stream temperature at a large

distance from the grain surface. The Granular Flow interface does not support the effect of fluid flow on a grain’s motion, so the **Convective Heat Transfer** feature should be used assuming there is fluid in the system, and that the fluid has zero effect on grain dynamics.

The heat transfer coefficient  $h$  can be specified directly or by entering the Nusselt number  $Nu$  (dimensionless),

$$Nu = \frac{d_g h}{k}$$

where  $k$  (SI unit:  $W/(m \cdot K)$ ) is the thermal conductivity of the fluid (assumed to be isotropic) and  $d_g$  (SI unit:  $m$ ) is the grain diameter.

### *Contact Search Theory*

---

DEM simulations often involve a large number of grains where, at any given time, each grain is in contact with a very small number of grains and wall elements. Therefore, an efficient contact search algorithm is required to determine the contact pairs. [The Granular Flow Interface](#) uses the no binary search (NBS) algorithm for detecting grain–grain contact pairs and the tree-based search algorithms for determining grain–wall contacts. The two algorithms are briefly described below. More details can be found in [Ref. 1](#).

#### **GRAIN–GRAIN CONTACT SEARCH**

The NBS algorithm consists of two steps: broad search and fine search. The broad search first organizes the grains into the grid cells based on their position. The fine search step then checks for potential contacts by surveying the grains in the adjacent cells.

##### *Broad Search*

The broad search begins by constructing a uniform rectangular grid across the entire geometry. The length of the grid cell in each direction is constrained to be larger than the maximum contact search radius. The contact search radius for each grain is equal to or greater than the grain radius depending on the **Contact search expansion ratio**  $\beta$  specified in the [Grain Properties](#) feature. Subsequently, if the total number of grid cells in each direction exceeds the **Maximum number of cells per direction** (specified in the physics interface settings), the grid cell length is adjusted to ensure that these limits are satisfied. Once the grid is constructed, each grain is indexed into a grid cell based on its position.

### *Fine Search*

Once all the grains have been assigned a cell index, the fine search step goes through each grain and checks for contact with every grain in its own cell and the neighboring cells. Note that the choice of the minimum length of the grid cell ensures that this step can be restricted to just the adjacent cells.

To avoid double counting the grain contacts, the fine search is restricted to a subset of neighboring cells. In 2D, if the own cell has the index (0, 0), then the neighboring cells that are searched for are indexed as

$$(-1, 0), (-1, -1), (0, -1), (1, -1)$$

Similarly in 3D, if the own cell has the index (0, 0, 0), then the neighboring cells that are searched for are indexed as

$$(0, 0, -1), (-1, 0, -1), (-1, 0, 0), (-1, 0, 1), (0, -1, -1), (0, -1, 0), (0, -1, 1) \\ (-1, -1, -1), (-1, -1, 0), (-1, -1, 1), (1, -1, -1), (1, -1, 0), (1, -1, 1)$$

Once a contact pair has been identified, the contact information is updated for both grains in the pair.

When a [Periodic Condition](#) feature is active in and the target cells are the are near the boundaries of the geometry, the neighboring cell list is modified to ensure that the contacts are detected with the periodic images of the grains from across the periodic boundaries.

When releasing grains, it is often necessary to ensure that the grain being considered for release has an acceptable overlap with preexisting grains. In such situations, every single neighboring cell is scanned (along with the own cell) for potential contacts.

### **GRAIN-WALL CONTACT SEARCH**

Tree based algorithms are used to detect grain-wall contacts. The wall elements are initially used to build a quadtree in 2D, or an octree in 3D, which are hierarchical data structures. The root node represents the entire space that is recursively divided into four quadrants in 2D or eight octants in 3D. Each wall element is inserted into the appropriate nodes based on their spatial location.

Once the tree is constructed, the search for a grain-wall contact consists of two stages. In the first stage, the tree is traversed beginning at the root node to identify the nodes of the tree associated with the spatial locations in the vicinity of the grain position. This step restricts the list of potential wall elements that need to be searched for contact.

The second stage then searches for contact between the grain and each of the previously identified wall elements.

### *Initial Conditions*

---

It is possible to release grains at user defined positions using the [Release from Grid](#) and [Release from Data File](#) features, or at arbitrary positions in the domains or on the boundaries using the [Release](#) and [Inlet](#) features, respectively.

When using the [Release from Grid](#) or [Release from Data File](#) features, the grains being released by a node must have the same material properties. These features are best suited for situations when the initial conditions of the grains to be released are predetermined.

The [Release](#) or [Inlet](#) features on the other hand allow for more flexibility both in terms of the initial positions and the material properties of the grains being released. The [Release](#) feature allows the initial positions to be either random or based on an analytic expression. The [Inlet](#) feature additionally supports a uniform distribution of the initial positions along the boundaries. Both the features support releasing multiple types of grains using a single node. The distribution of the grain types can be directly specified, or in terms of the number fraction, mass fraction, or volume fractions.

When grains are released using any of the four release features, the grains are released sequentially. If the grains are released using the [Release](#) or [Inlet](#) features, the species index of the grain is selected randomly according to the specified distribution. The grain is released at the release position only if the overlap with any grain or wall element is within acceptable limits. The contact search algorithm used to check for the overlap is described in [Contact Search Theory](#). It is possible to override the contact search when using the [Release from Grid](#) or [Release from Data File](#) features.

Apart from the initial positions of the grains, other parameters such as the initial velocities and angular velocities also need to be specified using expressions. Additionally, if the **Compute grain temperature** checkbox is enabled, the initial temperature of the grains also need to be specified.

### *Boundary Conditions*

---

The boundary conditions supported by the Granular Flow interface are controlled by the [Wall](#), [Inlet](#), [Outlet](#), and [Periodic Condition](#) features.

## WALL

The boundaries selected by the **Wall** feature are treated as rigid walls that can exert contact forces on the grains when they come into contact. The contact force models selected in the settings window of **The Granular Flow Interface** interface are used to model the grain–wall interactions. The corresponding contact parameters are specified in the **Contact with Walls** feature. Additionally, if the **Compute Grain Temperature** and the **Compute Conductive Heat Transfer** checkboxes are enabled, the boundary conditions for the wall temperature also need to be specified.

Many applications benefit from moving walls, and the motion of the walls is also controlled by the **Wall Movement** settings of this feature. The boundaries are treated as rigid walls, and no deformation is allowed. The allowed rigid body movements include translation, rotation, and a combination of the two.

## INLET

The **Inlet** feature is used to release grains along the boundary such that the grain centers lie on the boundary at release time. This feature overrides any **Wall** features that are located above it in the **Model Builder** window, and therefore the selected boundaries are not considered for grain–wall interactions.

## OUTLET

The **Outlet** feature allows the grains to pass through different adjoining domains or to remove the grains passing through it from the simulation domain. This feature naturally overrides the **Wall** features that are located above it in the **Model Builder** window, and therefore the selected boundaries are not considered for grain–wall interactions. This feature can be combined with the **Inlet** feature to affect the behavior of the grains passing through the boundaries selected in the **Inlet** feature.

## PERIODIC CONDITION

The **Periodic Condition** feature is used to apply periodic boundary conditions in one of the  $x$ ,  $y$ , or  $z$  directions. Grains near the periodic boundaries can interact with the periodic images of the grains and wall elements, and grains passing through the source boundaries will be replaced by their periodic images at the destination boundary.

This feature naturally overrides the **Wall** features that are located above it in the **Model Builder** window, and therefore the selected boundaries are not considered for grain–wall interactions. Additionally, this feature also overrides the **Outlet** feature but can be used along with the **Inlet** feature to release grains along the periodic boundaries.

## *Auxiliary Dependent Variables Theory*

---

In certain cases, it is desirable to add degrees of freedom to the grains. You can do so by using the [Auxiliary Dependent Variable](#) feature. The additional degrees of freedom are solved for each grain. For each degree of freedom,  $w$ , the following ODE is solved for each grain:

$$\frac{dw}{dt} = R$$

where  $R$  is a user-defined source term. When the **Integrate** option is set to **Along grain trajectory**, the following ODE is solved for each grain instead:

$$\frac{dw}{ds} = R$$

where  $s$  (SI unit: m) is the direction tangential to the motion of the grain.

## *Accumulator Theory: Domains*

---

The [Accumulator \(Domain\)](#) node is used to transfer information from grains to the domains they occupy or pass through. Each **Accumulator** defines a variable, called the accumulated variable, in the selected domains. The accumulated variable is discretized using constant shape functions, so its value is uniform over every mesh element and can be discontinuous between adjacent mesh elements.

The name of the accumulated variable is specified in the **Accumulated variable name** field in the **Accumulator Settings** section of the settings window. The default variable name, `rpd`, will be used in the remainder of this section when referring to the accumulated variable.

### **ACCUMULATOR TYPE**

The options in the **Accumulator type** list are **Density** and **Count**. If **Density** is selected, the source term is divided by the area or volume of the mesh element when calculating each grain's contribution to the accumulated variable. If **Count** is selected, no division by the area or volume of the mesh element occurs.

The equations in the following section are valid for the **Density** type. The corresponding value of the accumulated variable for the **Count** type is

$$\text{rpd}_{\text{count}} = \text{rpd}_{\text{density}} \times V$$

where  $V$  is the mesh element volume (in 3D) or area (in 2D).

## ACCUMULATION OVER ELEMENTS

When **Elements** is selected from the **Accumulate over** list, the value of the accumulated variable in a mesh element is the sum of the source terms  $R_i$  evaluated for all grains in that mesh element:

$$\text{rpd} = \frac{1}{V} \sum_{i=1}^N R_i$$

where  $N$  is the total number of grains in the element and  $V$  is the area (2D) or volume (3D) of the mesh element. In other words, the contribution of each grain to the accumulated variable is distributed uniformly over the mesh element the grain is in, regardless of the grain's exact position within the element.

If **Elements and time** is selected from the **Accumulate over** list, then the sum of the source terms for the grains in the mesh element is used to define the time derivative of the accumulated variable, rather than its instantaneous value:

$$\frac{d(\text{rpd})}{dt} = \frac{1}{V} \sum_{i=1}^N R_i$$

Thus, the value of the accumulated variable depends on the time history of the grains in the mesh element, instead of the instantaneous positions of the grains. As each grain propagates, it will leave behind a trail based on its contributions to the accumulated variables in the mesh elements it has traversed. The algorithm for accumulating over time takes into account the fraction of a time step taken by the solver that the grain spends in each mesh element, even if it crosses between elements during the time step.

### *Accumulator Theory: Boundaries*

---

The **Accumulator (Boundary)** feature transfers information from grains to the boundaries they hit. Each **Accumulator** defines a variable, called the accumulated variable, on the selected boundaries. The accumulated variable is discretized using constant shape functions, so its value is uniform over every mesh element and may be discontinuous between adjacent mesh elements.

The name of the accumulated variable is specified in the **Accumulated variable name** field in the **Accumulator Settings** section of the settings window. The default variable name, **rpb**, will be used in the remainder of this section when referring to the accumulated variable.

The options in the **Accumulator type** list are **Density** and **Count**. If **Density** is selected, the source term is divided by the surface area or length of the boundary mesh element when calculating each grain's contribution to the accumulated variable. If **Count** is selected, no division by the surface area or length of the boundary element occurs.

The equations in the following section are valid for the **Density** type. The corresponding value of the accumulated variable for the **Count** type is

$$\text{rpb}_{\text{count}} = \text{rpb}_{\text{density}} \times V$$

where  $V$  is the boundary element surface area (in 3D) or length (in 2D).

The accumulated variable in a boundary element gets incremented by the source term  $R$  whenever a grain hits the boundary:

$$\text{rpb}_{\text{new}} = \text{rpb} + \frac{R}{V}$$

where division by the mesh element area or length occurs because the accumulator is assumed to be of type **Density**. Thus the source term evaluated for an incident grain is uniformly distributed over the boundary element. It is possible for the same grain to increment the accumulated variable in many different boundary elements or even in the same element multiple times.

### BUILT-IN GLOBAL VARIABLES

By default, the boundary **Accumulator** defines the following global variables:

TABLE 3-1: BUILT-IN GLOBAL VARIABLES FOR BOUNDARY ACCUMULATORS.

NAME	DESCRIPTION
<scope>.<name>_int	Integral of accumulated variable
<scope>.<name>_ave	Average of accumulated variable
<scope>.<name>_max	Maximum of accumulated variable
<scope>.<name>_min	Minimum of accumulated variable
<scope>.<name>_sum	Sum of accumulated variable over elements

Here, <scope> includes the physics interface name and <name> the Accumulator and parent feature. For example, the average of the accumulated variable over a boundary may be called `gran.wp1.bacc1.rpb_ave`, where `gran` is the name of the Granular Flow interface, `wp1` is the name of the parent Wall node, `bacc1` is the name of the Accumulator node, and `rpb` is the accumulated variable name. These variables are all

available in the **Insert Expression** and **Replace Expression** menus during results evaluation.

These global variables are computed by defining a set of nonlocal couplings on the selection of the parent physics feature, such as the **Wall** feature to which the **Accumulator** is added. The following expressions for the global variables are used.

TABLE 3-2: BUILT-IN GLOBAL VARIABLE DEFINITIONS FOR BOUNDARY ACCUMULATORS.

NAME	EXPRESSION
<scope>.<name>_ave	<wscope>.ave(<scope>.<name>)
<scope>.<name>_int	<wscope>.sum(<scope>.<name>)
<scope>.<name>_max	<wscope>.max(<scope>.<name>)
<scope>.<name>_min	<wscope>.min(<scope>.<name>)
<scope>.<name>_sum	<wscope>.sum(<scope>.<name>/<scope>.meshVol)

Here, <wscope> is the scope of the parent boundary feature; for example, gran.wp1.

### *Force Accumulator Theory*

The **Force Accumulator** feature stores the normal component of contact force on the wall when a grain hits the wall. Each **Force Accumulator** feature defines a variable, called the accumulated variable, on the selected boundaries. The accumulated variable is discretized using constant shape functions, so its value is uniform over every mesh element and can be discontinuous between adjacent mesh elements.

The name of the accumulated variable is specified in the **Accumulated variable name** field in the **Accumulator Settings** section of the settings window. The default variable name, wf, will be used in the remainder of this section when referring to the accumulated variable. The accumulated variable in a boundary element gets incremented according to

$$wf_{\text{new}} = wf + |\mathbf{F}_n|$$

where  $\mathbf{F}_n$  is the normal contact force. For the theory on contact forces, see [Contact Force: Linear Elastic Model](#), [Contact Force: Hertz–MD \(Mindlin and Deresiewicz\) Model](#), and [Contact Force: Hertz–MD with Adhesion Model](#). It is possible for the same grain to increment the accumulated variable in many different boundary elements, or even in the same element multiple times.

Similar to the [Accumulator \(Boundary\)](#) feature, the **Force Accumulator** feature also defines global variables; for details, see the [Built-in Global Variables](#) subsection under [Accumulator Theory: Boundaries](#) and replace `rpb` with `wf`.

Unlike the [Accumulator \(Boundary\)](#) feature, the Settings window for the **Force Accumulator** feature does not include an **Accumulator type** list; the **Force Accumulator** feature is an **Accumulator** boundary feature of **Count** type. To calculate the accumulated force on a boundary element divided by the boundary element surface area (equivalent to the **Density** type of the **Accumulator** boundary feature), use the expression `<scope>.<name>/meshvol`, for example `gran.wp1.facc1.wf/meshvol`, in the results processing. See [Accumulator Theory: Boundaries](#) to learn more about the accumulator types **Count** and **Density**.

### *Time Step Size*

---

The Granular Flow interface suggests a time step for the model set up and stores it in a variable, `<scope>.ts` (for example, `gran.ts`). This variable is automatically used in the solver for the time step used. It should be noted that it is just a suggestion, and users are encouraged to verify the time step size and change the value or expression as needed. The expressions for the suggested time step depends on the contact force model used. When the **Linear elastic model** is selected in the **Contact force model** list in the physics interface node's **Force** section, the suggested time step is calculated as:

$$ts = 0.2 \sqrt{\frac{m_g}{k_n}} \quad (3-31)$$

where,

- $m_g$  is the mass of the grain
- $k_n$  is the normal spring coefficients

If model other than **Linear elastic** is selected, then the suggested time step is calculated based on the Rayleigh time step criteria ([Ref. 7](#)):

$$ts = 0.2 \frac{\pi r_g \sqrt{\frac{\rho_g}{G_g}}}{0.1631v_g + 0.8766} \quad (3-32)$$

- $r_g$  is the radius of the smallest grain
- $\rho_g$  is the value of the least dense grain

- $G_g$  is the maximum value of shear modulus
- $\nu_g$  is the maximum value of Poisson's ratio

Equation 3-32 is calculated for each grain species, and the smallest value is stored in the variable `<scope>.ts`.

The coefficient 0.2 in both Equation 3-31 and Equation 3-32 is used to make the time step less restrictive so that the simulation, in most cases, does not become unstable due to large time step issue.

It is to be noted that the variable `<scope>.ts` is a global variable as it needs to be accessible to the solver, so it can only be computed using the properties of grains that are global variables as well. The properties of walls cannot be used directly as the variables are defined on the boundaries. Users are suggested to manually calculate the suggested time step based on the above equations if wall material properties are to be used.

## References

---

1. H.R. Norouzi, R. Zarghami, R. Sotudeh-Gharebagh, and N. Mostoufi, *Coupled CFD-DEM Modeling: Formulation, Implementation and Application to Multiphase Flows*, John Wiley & Sons, 2016.
2. K.L. Johnson, *Contact Mechanics*, Cambridge University Press, 1985.
3. A. Chokshi, A.G.G.M. Tielens, and D. Hollenbach, "Dust Coagulation," *Astrophys. J.*, vol. 407, pp. 806–819, 1993.
4. E.J. Parteli, J. Schmidt, C. Blümel, K.E. Wirth, W. Peukert, and T. Pöschel, "Attractive particle interaction forces and packing density of fine glass powders," *Sci. Rep.*, vol. 4, 6227, 2014.
5. J. Ai, J.F. Chen, J.M. Rotter, and J.Y. Ooi, "Assessment of rolling resistance models in discrete element simulations," *Powder Technol.*, vol. 206, pp. 269–282, 2011.
6. Z. Peng, E. Doroodchi, and B. Moghtaderi, "Heat transfer modelling in discrete element method (DEM)-based simulations of thermal process: theory and model development," *Prog. Energy Combust. Sci.*, vol. 79, 100847, 2020.
7. S.B. Yeom, E.S. Ha, M.S. Kim, S.H. Jeong, S.J. Hwang, and D.H. Choi, "Application of the Discrete Element Method for Manufacturing Process Simulation in the Pharmaceutical Industry," *Pharmaceutics*, vol. 11, no. 8, p. 414, 2019.

# I n d e x

- A** accumulator (node), boundaries 53, 55
  - accumulator (node), domains 55
  - accumulator theory, boundaries 78
  - accumulator theory, domains 77
  - Application Libraries window 9
  - application library examples
    - powder spreading 20
    - ribbon mixer 15
  - auxiliary dependent variable 77
  - auxiliary dependent variable (node) 49
- C** common settings 6
  - contact between grains (node) 31
  - contact with walls (node) 34
  - convective heat losses (node) 52
- D** documentation 7
- E** emailing COMSOL 9
- F** force (node) 48
- G** grain properties (node) 27
  - Granular Flow interface 22
- I** inlet (node)
  - particle tracing 43
  - internet resources 7
- K** knowledge base, COMSOL 10
- M** MPH files 9
- N** nodes, common settings 6
- O** outlet (node) 48
- P** physics interfaces, common settings 6
- R** release (node) 39
  - release from data file (node) 46
  - release from grid (node) 43
- S** standard settings 6
- T** technical support, COMSOL 9
  - temperature calculation 70
  - torque (node) 48
  - total number of particles 16
- W** wall (node) 29
  - websites, COMSOL 10

