

# Particle Tracing Module

User's Guide

# Particle Tracing Module User's Guide

© 1998–2017 COMSOL

Protected by U.S. Patents listed on [www.comsol.com/patents](http://www.comsol.com/patents), and U.S. Patents 7,519,518; 7,596,474; 7,623,991; 8,457,932; 8,954,302; 9,098,106; 9,146,652; 9,323,503; 9,372,673; and 9,454,625. Patents pending.

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

COMSOL, the COMSOL logo, COMSOL Multiphysics, Capture the Concept, COMSOL Desktop, LiveLink, and COMSOL Server 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 5.3

## Contact Information

Visit the Contact COMSOL page at [www.comsol.com/contact](http://www.comsol.com/contact) to submit general inquiries, contact Technical Support, 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 at the COMSOL Access page at [www.comsol.com/support/case](http://www.comsol.com/support/case). Other useful links include:

- 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/support/updates](http://www.comsol.com/support/updates)
- COMSOL Blog: [www.comsol.com/blogs](http://www.comsol.com/blogs)
- Discussion Forum: [www.comsol.com/community](http://www.comsol.com/community)
- Events: [www.comsol.com/events](http://www.comsol.com/events)
- COMSOL Video Gallery: [www.comsol.com/video](http://www.comsol.com/video)
- Support Knowledge Base: [www.comsol.com/support/knowledgebase](http://www.comsol.com/support/knowledgebase)

Part number: CM022701

# C o n t e n t s

## Chapter 1: Introduction

<b>About the Particle Tracing Module</b>	<b>10</b>
What Can the Particle Tracing Module Do? . . . . .	10
Particle Tracing Plots, Data Sets, Derived Values, and Studies . . . . .	11
The env and bndenv Operators . . . . .	12
The Particle Tracing Module Physics Interface Guide . . . . .	13
Common Physics Interface and Feature Settings and Nodes. . . . .	14
Where Do I Access the Documentation and Application Libraries? . . . . .	15
<b>Overview of the User's Guide</b>	<b>18</b>

## Chapter 2: Particle Tracing Modeling

<b>Particle Tracing</b>	<b>20</b>
Introduction to Particle Tracing Modeling . . . . .	20
Charged Particle Tracing . . . . .	21
Particle Tracing for Fluid Flow . . . . .	23
Mathematical Particle Tracing . . . . .	30
Multiphysics Interfaces. . . . .	30
<b>Modeling Tools</b>	<b>32</b>
Choosing a Formulation . . . . .	32
Geometry and Mesh Settings . . . . .	36
Special Variables . . . . .	38
Component Couplings. . . . .	42
Monte Carlo Modeling. . . . .	44
Study Setup . . . . .	46
Auxiliary Dependent Variables and Residence Time. . . . .	48
Initialization of Auxiliary Dependent Variables . . . . .	50
Accumulators . . . . .	51
Overriding Particle Properties . . . . .	52

Filters . . . . .	53
Particle Counters . . . . .	54
Improving Plot Quality . . . . .	55
References . . . . .	57

## Chapter 3: Mathematical Particle Tracing

<b>The Mathematical Particle Tracing Interface</b>	<b>60</b>
Domain, Boundary, Pair, and Global Nodes for the Mathematical Particle Tracing Interface . . . . .	65
Wall. . . . .	67
Periodic Condition . . . . .	70
Accumulator (Boundary) . . . . .	71
Particle Counter . . . . .	73
Secondary Emission. . . . .	73
Particle Properties . . . . .	77
Force . . . . .	79
Rotating Frame . . . . .	80
Velocity Reinitialization . . . . .	81
Accumulator (Domain) . . . . .	82
Particle-Particle Interaction . . . . .	84
Release . . . . .	86
Axial Symmetry . . . . .	93
Inlet. . . . .	94
Nonlocal Accumulator. . . . .	98
Outlet. . . . .	99
Particle Continuity . . . . .	100
Auxiliary Dependent Variable . . . . .	100
Release from Edge . . . . .	101
Release from Point . . . . .	101
Release from Grid . . . . .	102
Override Properties . . . . .	103
Release from Data File. . . . .	103
<b>Theory for the Mathematical Particle Tracing Interface</b>	<b>106</b>
Newtonian Formulation . . . . .	106

Hamiltonian Formulation . . . . .	108
Lagrangian Formulation . . . . .	108
Massless Formulation . . . . .	109
Particle Tracing in Rotating Frames . . . . .	109
Initial Conditions — Position . . . . .	110
Initial Conditions — Velocity . . . . .	111
Particle-Particle Interactions . . . . .	113
Auxiliary Dependent Variables. . . . .	116
About the Boundary Conditions for the Particle Tracing Interfaces . . . . .	116
Accumulator Theory: Domains . . . . .	119
Accumulator Theory: Boundaries . . . . .	120
References for the Mathematical Particle Tracing Interface . . . . .	122

## Chapter 4: Charged Particle Tracing

<b>The Charged Particle Tracing Interface</b> . . . . .	<b>124</b>
Domain, Boundary, Pair, and Global Nodes for the Charged Particle Tracing Interface. . . . .	127
Collisions. . . . .	128
Elastic . . . . .	130
Excitation . . . . .	132
Attachment . . . . .	132
Ionization. . . . .	133
Resonant Charge Exchange . . . . .	134
Nonresonant Charge Exchange . . . . .	135
User Defined . . . . .	136
Friction Force . . . . .	137
Particle-Matter Interactions. . . . .	138
Ionization Loss. . . . .	139
Nuclear Stopping . . . . .	140
Particle Beam . . . . .	141
Thermionic Emission . . . . .	151
Particle Properties . . . . .	153
Symmetry . . . . .	154
Electric Force . . . . .	154
Magnetic Force . . . . .	156

Space Charge Density Calculation . . . . .	158
Surface Charge Density . . . . .	159
Current Density . . . . .	159
Heat Source. . . . .	160
Etch. . . . .	160
<b>Theory for the Charged Particle Tracing Interface</b>	<b>162</b>
Introduction to the Charged Particle Tracing Interface Theory . . . . .	162
Electric Force Theory . . . . .	164
Magnetic Force Theory . . . . .	164
Collisional Force Theory . . . . .	165
Particle-Matter Interaction Theory . . . . .	169
Particle Beam Theory . . . . .	172
Thermionic Emission Theory . . . . .	178
Surface Charge Density Theory . . . . .	183
Current Density Theory . . . . .	183
Heat Source Theory . . . . .	184
Etch Theory. . . . .	185
References for the Charged Particle Tracing Interface . . . . .	186

## Chapter 5: Particle Tracing for Fluid Flow

<b>The Particle Tracing for Fluid Flow Interface</b>	<b>188</b>
Domain, Boundary, Pair, and Global Nodes for the Particle Tracing for Fluid Flow Interface . . . . .	192
Particle Properties . . . . .	194
Symmetry . . . . .	195
Drag Force . . . . .	195
Lift Force. . . . .	199
Brownian Force . . . . .	200
Gravity Force . . . . .	202
Acoustophoretic Force . . . . .	203
Electric Force . . . . .	204
Magnetic Force . . . . .	205
Dielectrophoretic Force . . . . .	205
Shell. . . . .	207

Magnetophoretic Force . . . . .	208
Thermophoretic Force . . . . .	209
Erosion . . . . .	211
Mass Deposition . . . . .	212
Boundary Load . . . . .	213
Mass Flux. . . . .	213
Volume Force Calculation . . . . .	213
Convective Heat Losses . . . . .	215
Radiative Heat Losses . . . . .	215
Heat Source. . . . .	216
Droplet Breakup . . . . .	216
Kelvin-Helmholtz Breakup Model. . . . .	217
Rayleigh-Taylor Breakup Model . . . . .	218
Nozzle. . . . .	218
Nozzle Domain . . . . .	221
<b>The Droplet Sprays in Fluid Flow Interface</b>	<b>222</b>
<b>Theory for the Particle Tracing for Fluid Flow Interface</b>	<b>223</b>
Particle Motion in a Fluid. . . . .	223
Particle Motion in a Turbulent Flow. . . . .	230
Particle Motion in a Shear Flow . . . . .	236
Additional Forces. . . . .	240
Erosion Theory . . . . .	249
Droplet Breakup Theory. . . . .	251
Nozzle Theory . . . . .	255
Computing Particle Temperature. . . . .	257
Computing Particle Mass . . . . .	259
References for the Particle Tracing for Fluid Flow Interface . . . . .	259

## Chapter 6: Multiphysics Interfaces

<b>The Particle Field Interaction, Non-Relativistic Interface</b>	<b>264</b>
Electric Particle Field Interaction . . . . .	266
Space Charge Limited Emission . . . . .	268

<b>Theory for the Particle Field Interaction, Non-Relativistic Interface</b>	<b>270</b>
Electrostatics and Charged Particle Tracing Equations . . . . .	270
Space Charge Density Calculation . . . . .	271
Stabilization of the Space Charge Density Calculation . . . . .	275
Theory for the Space Charge Limited Emission Node . . . . .	276
<b>The Particle Field Interaction, Relativistic Interface</b>	<b>280</b>
Magnetic Particle Field Interaction . . . . .	283
<b>Theory for the Magnetic Particle Field Interaction, Relativistic Interface</b>	<b>285</b>
Electrostatics and Charged Particle Tracing Equations . . . . .	285
Current Density Calculation . . . . .	287
<b>The Fluid-Particle Interaction Interface</b>	<b>291</b>
Fluid-Particle Interaction . . . . .	293
<b>Theory for the Fluid-Particle Interaction Interface</b>	<b>295</b>
Laminar Flow and Particle Tracing Equations . . . . .	295
Volume Force Calculation . . . . .	296

## Chapter 7: Glossary

<b>Glossary of Terms</b>	<b>302</b>
--------------------------	------------



# Introduction

This guide describes the Particle Tracing Module, an optional add-on package for COMSOL Multiphysics® designed to compute particle trajectories. The particles can interact with boundaries and their motion can be affected by external fields, which may be user-defined or solved for by other physics interfaces.

This chapter introduces you to the capabilities of this module. A summary of the physics interfaces and where you can find documentation and model examples is also included. The last section is a brief overview with links to each chapter in this guide.

In this chapter:

- [About the Particle Tracing Module](#)
- [Overview of the User's Guide](#)

# About the Particle Tracing Module

These topics are included in this section:

- [What Can the Particle Tracing Module Do?](#)
- [Particle Tracing Plots, Data Sets, Derived Values, and Studies](#)
- [The env and bndenv Operators](#)
- [The Particle Tracing Module Physics Interface Guide](#)
- [Where Do I Access the Documentation and Application Libraries?](#)



[The Physics Interfaces and Building a COMSOL Multiphysics Model in the COMSOL Multiphysics Reference Manual](#)

---

## *What Can the Particle Tracing Module Do?*

---

The Particle Tracing Module is a general purpose, flexible tool that is used to compute the trajectories of particles. The particles can be affected by various forces, particle-boundary interactions, and particle-domain interactions. One-way and two-way couplings between particles and external fields are also supported.

A custom physics interface is available for modeling the motion of electrons and ions in electromagnetic fields. This makes it possible to model devices such as magnetic lenses, electron guns, and mass spectrometers. There are a number of tools available to extract typical quantities of interest and to make plots of the particle trajectories, including phase portraits and Poincaré maps. Built-in multiphysics interfaces are available to model the interaction of particles with electric and magnetic fields. Dedicated features are available to perform Monte Carlo simulations of the collisions between ions or electrons and the molecules in a rarefied gas.

There is also a dedicated physics interface for computing particle trajectories in a fluid system. There is a wide variety of predefined forces, including drag, gravity, electric, and Brownian force. Fluid-based particle tracing allows for detailed investigation of mixing, separation, and filtering devices. Furthermore, built-in features are available for modeling complicated physical processes such as thermophoresis, dielectrophoresis, magnetophoresis, and acoustophoresis.

A mathematical particle tracing interface provides complete freedom and flexibility to define the equations of motion governing particle trajectories. In addition to built-in models to describe Newton's law of motion, there are dedicated formulations available to specify the Lagrangian or Hamiltonian that dictates the motion of the particles.

All of the physics interfaces provide many different ways of releasing particles. Particles can be released from geometric entities of any level, including domains, boundaries, edges, and points. They can also be released by specifying the initial coordinates directly or by importing sets of initial coordinates from a file. Particles can also be released at multiple different times. There are many options available for describing how the particles interact with boundaries, including diffuse and specular reflection, sticking probabilities, and secondary particle emission.

The physics interfaces can all be used with COMSOL Multiphysics, but when tracing charged particles the AC/DC Module is often beneficial because it can be used to model complex alternating or direct current systems and includes more options for computing magnetic fields. If the particles move through a rarefied gas, as is often the case for ion and electron beams in vacuum systems, the Molecular Flow Module is useful because it provides built-in tools to compute the density of extremely rarefied gases. Similarly, when tracing particles in a fluid system the CFD Module or Microfluidics Module is often useful due to the multitude of advanced fluid flow features available in those modules.

### *Particle Tracing Plots, Data Sets, Derived Values, and Studies*

---

The Particle Tracing Module uses some operators, studies, and postprocessing features that are only available with this module. The descriptions for these features are, however, described with the generic plot, data set, and study types. Go to the links below for more information.

In the *COMSOL Multiphysics Reference Manual*:

- [Particle Trajectories](#) (study)
- [Bidirectionally Coupled Particle Tracing](#) (study)
- [Particle \(Data Set\)](#) (data set)
- [Particle Trajectories](#) (plot) and [Filter for Particle Trajectories](#)
- [Phase Portrait](#) (plot)
- [Poincaré Map](#) (plot)
- [Particle Evaluation](#)



---

The links to the nodes described in the *COMSOL Multiphysics Reference Manual* do not work in the PDF, only from the on line help in COMSOL Multiphysics.

---



For an example of:

- [Particle Trajectories](#) see *Ion Cyclotron Motion*: Application Library path: **Particle\_Tracing\_Module/Charge\_Particle\_Tracing/ion\_cyclotron\_motion**.
  - [Poincaré Map](#) see *Rössler Attractor*: Application Library path **Particle\_Tracing\_Module/Tutorials/rossler\_attractor**.
  - [Phase Portrait](#) see *Ideal Cloak*: Application Library path **Particle\_Tracing\_Module/Ray\_Tracing/ideal\_cloak**.
- 

### *The env and bndenv Operators*

---



The Particle Tracing Module has these operators that are only available with this module. See [Built-In Operators](#) in the *COMSOL Multiphysics Reference Manual* for additional information.

---

- Evaluating `env(expr)` on a particle, evaluates the expression `expr` at the point in the domain where the particle is. When evaluating a variable `var` on a particle, if the variable is not defined on the particle it is automatically replaced by `env(var)`. Therefore the `env` operator can often be omitted. If `env(var)` is evaluated when a particle is on a boundary, and `var` is defined on two domains adjacent to the boundary, the operator returns the arithmetic mean of `var` on both sides of the boundary.
- Evaluating `bndenv(expr)` on a particle, evaluates `expr` at the point on the boundary where the particle is. If the particle is not on a boundary, the evaluation fails. Use this operator instead of `env` when evaluating expressions that are only defined on boundaries.
- The operators `env_in(expr)` and `env_out(expr)` can only be used when a particle is in contact with a boundary. These operators evaluate the expression `expr` in the domain the particle reaches the boundary from and the domain on the opposite side

of the boundary, respectively. These operators are often used to evaluate an expression that changes discontinuously at the boundary.

- The operators `env_re1(expr)` and `bdenv_re1(expr)` are analogous to `env(expr)` and `bdenv(expr)`, respectively, except that they always evaluate expressions at the initial particle positions.

### *The Particle Tracing Module Physics Interface Guide*









The Particle Tracing 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 Particle Tracing 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*:

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

PHYSICS INTERFACE	ICON	TAG	SPACE DIMENSION	AVAILABLE PRESET STUDY TYPE
<b>AC/DC</b>				
<b>Particle Tracing</b>				
Particle Field Interaction, Non-Relativistic		—	3D, 2D, 2D axisymmetric	bidirectionally coupled particle tracing; time dependent
Particle Field Interaction, Relativistic (requires the AC/DC Module and Particle Tracing Module)		—	3D, 2D, 2D axisymmetric	bidirectionally coupled particle tracing; time dependent

PHYSICS INTERFACE	ICON	TAG	SPACE DIMENSION	AVAILABLE PRESET STUDY TYPE
Charged Particle Tracing		cpt	3D, 2D, 2D axisymmetric	bidirectionally coupled particle tracing; time dependent
 <b>Fluid Flow</b>				
 <b>Particle Tracing</b>				
Particle Tracing for Fluid Flow		fpt	3D, 2D, 2D axisymmetric	bidirectionally coupled particle tracing; time dependent
Fluid-Particle Interaction		—	3D, 2D, 2D axisymmetric	bidirectionally coupled particle tracing; time dependent
Droplet Sprays in Fluid Flow		—	3D, 2D, 2D axisymmetric	bidirectionally coupled particle tracing; time dependent
 <b>Mathematics</b>				
Mathematical Particle Tracing		pt	3D, 2D, 2D axisymmetric	bidirectionally coupled particle tracing; time dependent

### *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 2-3](#) for links to common sections and [Table 2-4](#) 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 internet 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 (as long as you have a license), 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 about 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 many of the features on the GUI. 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*.

---


To open the **Help** window:


- In the **Model Builder**, **Application Builder**, or **Physics Builder** click a node or window and then press F1.
- On any toolbar (for example, **Home**, **Definitions**, or **Geometry**), hover the mouse over a button (for example, **Add Physics** or **Build All**) and then press F1.
- From the **File** menu, click **Help** (  ).
- In the upper-right corner of the COMSOL Desktop, click the **Help** (  ) button.

Win

Mac	<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> </ul>
Linux	<ul style="list-style-type: none"> <li>• On 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*

Win	<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>
-----	--

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

**THE APPLICATION LIBRARIES WINDOW**

Each application includes documentation with the theoretical background and step-by-step instructions to create a model application. The applications are available in COMSOL as MPH-files that you can open for further investigation. You can use the step-by-step instructions and the actual applications as a template for your own modeling and applications. In most models, SI units are used to describe the relevant properties, parameters, and dimensions in most examples, 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 application and its properties, including options to open it or a PDF document.







	<p><a href="#">The Application Libraries Window</a> in the <i>COMSOL Multiphysics Reference Manual</i>.</p>
---	---



### Opening the Application Libraries Window

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

---

	<ul style="list-style-type: none"><li>• From the <b>Home</b> toolbar, <b>Windows</b> menu, click (  ) <b>Applications Libraries</b>.</li><li>• From the <b>File</b> menu select <b>Application Libraries</b>.</li></ul> <p>To include the latest versions of model examples, from the <b>File&gt;Help</b> menu, select (  ) <b>Update COMSOL Application Library</b>.</p>
	Select <b>Application Libraries</b> from the main <b>File&gt;</b> or <b>Windows&gt;</b> menus.
	To include the latest versions of model examples, from the <b>Help</b> menu select (  ) <b>Update COMSOL Application Library</b> .

---

### CONTACTING COMSOL BY EMAIL

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

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 are sent to you by email.

### 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>
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/support/updates">www.comsol.com/support/updates</a>
COMSOL Blog	<a href="http://www.comsol.com/blogs">www.comsol.com/blogs</a>
Discussion Forum	<a href="http://www.comsol.com/community">www.comsol.com/community</a>
Events	<a href="http://www.comsol.com/events">www.comsol.com/events</a>
COMSOL Video Gallery	<a href="http://www.comsol.com/video">www.comsol.com/video</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 *Particle Tracing Module User's Guide* gets you started with modeling 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.

## TABLE OF CONTENTS, GLOSSARY, AND INDEX

To help you navigate through this guide, see the [Contents](#), [Glossary of Terms](#), and [Index](#).

## PARTICLE TRACING MODELING

In [Particle Tracing Modeling](#), the different particle tracing interfaces are discussed (Charged Particle Tracing, Particle Tracing in Fluid Flow, and Mathematical Particle Tracing). In the [Modeling Tools](#) section, special variables, Monte Carlo modeling, filters, and auxiliary dependent variables and residence time are discussed.

## MATHEMATICAL PARTICLE TRACING

[The Mathematical Particle Tracing Interface](#) and its underlying theory are described in this chapter.

## CHARGED PARTICLE TRACING

[The Charged Particle Tracing Interface](#) and its underlying theory are described in this chapter.

## PARTICLE TRACING FOR FLUID FLOW

[The Particle Tracing for Fluid Flow Interface](#), [The Droplet Sprays in Fluid Flow Interface](#), and their underlying theory are described in this chapter.

## MULTIPHYSICS INTERFACES

The built-in multiphysics interfaces, including [The Particle Field Interaction](#), [Non-Relativistic Interface](#), [The Particle Field Interaction](#), [Relativistic Interface](#), and the [The Fluid-Particle Interaction Interface](#), are described in this chapter.

# Particle Tracing Modeling

This chapter gives an overview of the physics interfaces available for modeling the trajectories of particles and provides guidance on choosing the appropriate physics interface for a specific problem.

In this chapter:

- [Particle Tracing](#)
- [Modeling Tools](#)

# Particle Tracing

In this section:

- [Introduction to Particle Tracing Modeling](#)
- [Charged Particle Tracing](#)
- [Particle Tracing for Fluid Flow](#)
- [Mathematical Particle Tracing](#)

## *Introduction to Particle Tracing Modeling*

---

Particle tracing provides a Lagrangian description of a problem by solving ordinary differential equations using Newton's law of motion. The trajectories of individual particles are always solved for in the time domain. Newton's law of motion requires specification of the particle mass and all forces acting on the particle. The forces can be divided into two categories, those due to external fields and due to interactions between particles. Forces due to external fields are typically computed from a finite element model, using the physics interfaces available in COMSOL Multiphysics and its other optional add-on modules.

For each particle, a second-order ordinary differential equation is solved for each component of the position vector. This means that three ordinary differential equations are solved for each particle in 3D and two in 2D. A first-order formulation of Newton's law of motion is also available, in which coupled first-order ordinary differential equations are solved for the components of the particle position and velocity. The Mathematical Particle Tracing interface also includes a Hamiltonian formulation that solves coupled first-order ordinary differential equations for the particle position and generalized momentum.

At each time step taken by the solver, the forces acting on each particle are queried from the external fields at the current particle position. If particle-particle interaction forces are included in the model then they are added to the total force. The particle position is then updated and the process repeats until the specified end time for the simulation is reached. During each time step, the particles may interact with boundaries in the geometry, or they may be subjected to other phenomena that can discontinuously change the particle velocity.

Because the Particle Tracing Module uses a very general formulation for computing particle trajectories, the particle tracing interfaces can be used to model charged

particle motion in electromagnetic fields, large scale planetary and galactic movement, and particle motion in laminar, turbulent, and multiphase fluid systems.

### *Charged Particle Tracing*

---

The [Charged Particle Tracing Interface](#) is designed to model the motion of electrons, individual ions, or small ion clusters in electric and magnetic fields.

#### **SPACE CHARGE EFFECTS**

A computationally efficient way to approach charged particle tracing modeling is to first solve for any external electric or magnetic fields, using for example the [Electrostatics interface](#) or the [Magnetic Fields interface](#), and then use these fields to exert electric and magnetic forces on the ions or electrons via a unidirectional, or one-way, coupling. In this approach, the effect of the particles on the fields is neglected. The external fields can be solved for using a Stationary, Frequency Domain, or Time Dependent study. The particle trajectories are then solved for in a separate Time Dependent study. The fact that the particle trajectories are computed in their own study allows efficient and computationally inexpensive iterative solvers to be used.

If the number density of the charged particles is sufficiently large, the particles may begin to cause significant perturbations in the external fields. There is no specific magnitude of the number density at which the space charge effects become significant; rather, as a general rule they should be included if their contributions to the external electric or magnetic fields is of a comparable order of magnitude to the sources or boundary conditions included in the other physics, such as surfaces maintained at specified potentials or external current sources. The bidirectional, or two-way, coupling between particles and fields can be included in the model by using [The Particle Field Interaction, Non-Relativistic Interface](#), which automatically adds the dedicated [Electric Particle Field Interaction](#) Multiphysics node to account for space charge effects.

If, in addition, the particles move at relativistic speeds, the current density due to particle motion may become significant. [The Particle Field Interaction, Relativistic Interface](#) automatically adds the [Electric Particle Field Interaction](#) and [Magnetic Particle Field Interaction](#) Multiphysics nodes to account for the space charge density and current density of particles, respectively. Magnetic particle-field interactions are usually negligibly small compared to electric particle-field interactions at non-relativistic speeds.



---

*Electron Beam Divergence Due to Self Potential*: Application Library path **Particle\_Tracing\_Module/Charged\_Particle\_Tracing/electron\_beam\_divergence**

*Relativistic Paraxial Electron Beam*: Application Library path **Particle\_Tracing\_Module/Charged\_Particle\_Tracing/electron\_beam\_divergence\_relativistic**

---

The computational requirements for models that include particle-field interactions increase significantly over those that neglect them. If the fields and particle trajectories are directly coupled to each other, both must be computed in the same Time Dependent study, and a fairly small time step must be taken by the solver to account for the constantly changing electric potential. In addition, the space charge density and current density are computed using variables that are constant over each mesh element, so it may be necessary to refine the mesh or increase the number of model particles to more accurately model particle-field interactions.

If the fields are stationary, as often occurs when beams of particles are released at constant current, it is possible to significantly reduce the computational cost of the model by using a Stationary solver to compute the fields and a Time Dependent solver to compute the particle trajectories. It is also possible to create a solver loop that alternates between the Stationary and Time Dependent solvers so that a bidirectional coupling between the trajectories and fields can be established; a dedicated Bidirectionally Coupled Particle Tracing study step is available for setting up such a solver loop. The process of combining these solvers is described in the section [Study Setup](#).

### **COULOMB FORCES**

If the density of charged particles is extremely high then it can be necessary to include the Coulomb force that acts between the particles. This is done by adding a Particle-Particle Interaction node to the model. When particle-particle interactions are included in a model the computational requirements increase and scale as the number of particles squared. When including the Coulomb force, it is often best to start with a small number of particles, solve the model, and then assess whether or not the effect is important to the model.



---

Coulomb forces and bidirectionally coupled particle-field interactions usually shouldn't be included together in the same model because this causes the force exerted by each charged particle on other charged particles to be double-counted, once directly and once via the electric field defined in the modeling domain.

---

### *Particle Tracing for Fluid Flow*

---

The [Particle Tracing for Fluid Flow Interface](#) is designed for modeling microscopic and macroscopic particles in a background fluid. There are two phases in the system: a particle phase consisting of discrete bubbles, particles, droplets, and so forth, and a continuous phase in which the particles are immersed. In order for the particle tracing approach to be valid, the fluid system should be a *dilute* or *dispersed flow*. This means that the volume fraction of the particles is much smaller than the volume fraction of the continuous phase, generally less than 1%. When the volume fraction of the particles is not small, the fluid system is categorized as a *dense flow* and a different modeling approach is required.

It is important to realize that with the particle tracing approach, particles do not displace the fluid they occupy. In addition, the finite size of the particle is not taken into account when modeling particle-wall interactions. In other words, for purposes of detecting particle-boundary interactions, the particles are treated as point masses. The specification of particle diameter is mostly used for size-dependent forces, such as the drag and dielectrophoretic forces.

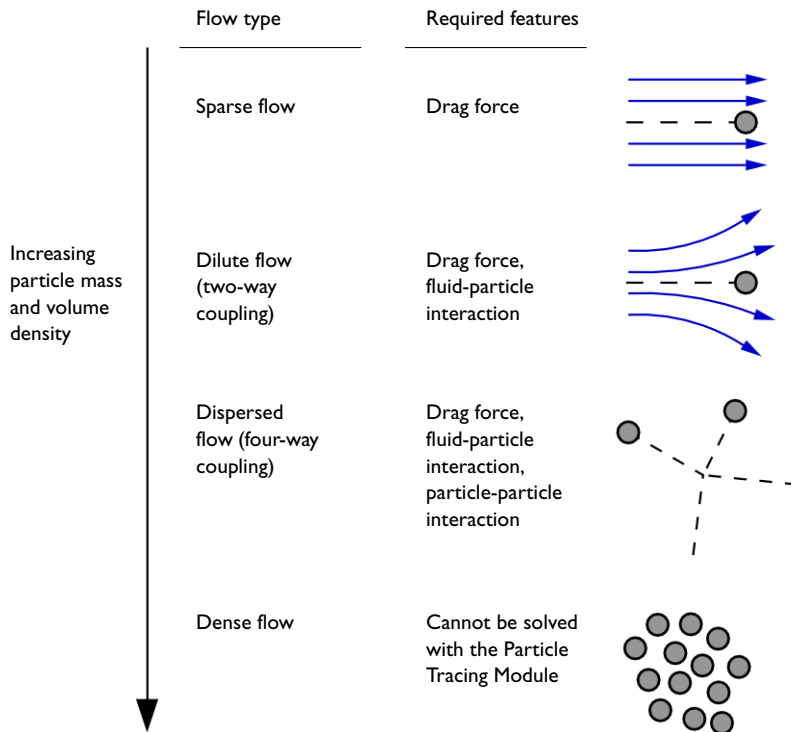
#### **SPARSE FLOW**

In a *sparse flow*, the continuous phase affects the motion of the particles but not vice-versa. This is often referred to as “one-way coupling.” When modeling such a system in COMSOL, it is usually most efficient to solve for the continuous phase and the dispersed phase in separate studies. The fluid usually affects the particle motion through the [Drag Force](#) feature, which defines a drag force based on the fluid properties, particle properties, and the particle velocity relative to the flow. Several built-in drag laws can be used, and the optimal drag law usually depends on the size and speed of the particles. Built-in options to apply random perturbations to account for turbulence in the fluid are also available.

For example, in the following model example, the velocity field is first computed using a Stationary study, then the particle trajectories are computed using a separate Time Dependent study.



*Particle Trajectories in a Laminar Static Mixer:* Application Library  
 path **Particle\_Tracing\_Module/Fluid\_Flow/laminar\_mixer\_particle**



*Figure 2-1: The physics features required to model sparse, dilute, and dispersed flows.*

### DILUTE FLOW

In a dilute flow the continuous phase affects the motion of the particles and the particle motion in turn disrupts the continuous phase. This is often referred to as a bidirectional coupling or “two-way coupling.” The bidirectional coupling between particles and fluids can be modeled using the [Fluid-Particle Interaction](#) Multiphysics node. This node can be added manually if the necessary physics interfaces are already



present. Alternatively, [The Fluid-Particle Interaction Interface](#) can be used to automatically add the necessary physics interfaces and Multiphysics nodes.

The body force exerted on the fluid by the particle is applied in an approximate way, in that it is smeared out over a mesh element. This smearing effect makes the volume force computed by the Fluid-Particle Interaction node somewhat mesh dependent. When modeling fluid-particle interactions for which the mass flow rate of particles is not constant, the continuous phase and dispersed phase must be computed simultaneously in the same study. The computational demand is significantly higher than in the [Sparse Flow](#) case.

If the fields are stationary, as often occurs when particles are released at constant mass flow rate, it may be possible to compute the particle trajectories using a Time Dependent solver while computing the fluid flow variables using a Stationary solver. It is also possible to create a solver loop that alternates between the Stationary and Time Dependent solvers so that a bidirectional coupling between the trajectories and fields can be established; a dedicated Bidirectionally Coupled Particle Tracing study step is available for setting up such a solver loop. The process of combining these solvers is described in the section [Study Setup](#).

## DISPERSED FLOW

In addition to the effects mentioned above, particle-particle interactions also need to be taken into account. This is often referred to as “four-way coupling.” Particle-particle interactions can be included in a model by adding a [Particle-Particle Interaction](#) node.

The following limitations apply:

- Hard sphere collisions are not supported. Forces must vary continuously with respect to the distance between particles as in, for example, the Coulomb force between charged particles.
- The computation time scales as the number of particles squared. This is because every particle interacts with every other particle over all distances. Although it is possible to apply a cut-off radius beyond which particles do not interact with each other, the interaction force must still be evaluated for every pair of particles.
- If the particle-particle interaction law is highly nonlinear, it can be necessary to use a very small time step. This is particularly true if the **Lennard-Jones** option is selected.
- The Particle-Particle Interaction feature produces a Jacobian matrix that is completely full. For a large number of particles, this is very expensive to factorize. By default, the **Exclude Jacobian contribution for particle-particle interaction** check

box is selected, which preserves the sparseness in the Jacobian. Clearing this check box is likely to result in a dramatic increase in the amount of memory and time needed to solve the problem.

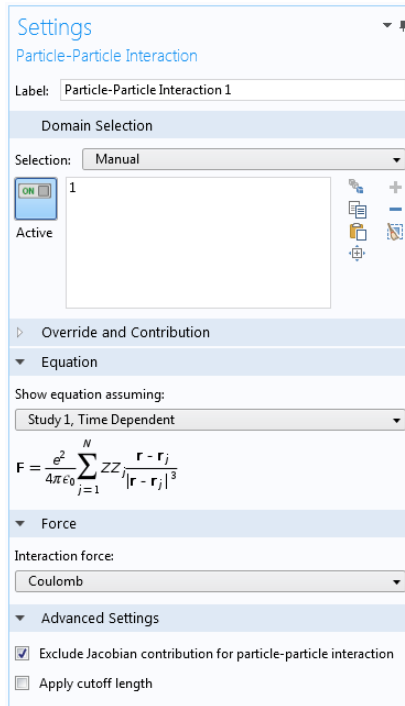


Figure 2-2: The Settings window for Particle-Particle Interaction.

The Particle Tracing for Fluid Flow interface provides a convenient way of computing the particle mass and temperature. Built-in auxiliary dependent variables for the mass and temperature can be activated by selecting the **Compute particle mass** and **Compute particle temperature** check boxes, respectively, in the physics interface **Additional Variables** section. When the option to compute particle mass is activated, it is possible to set the initial particle mass in particle release features, such as [Release](#), [Inlet](#), and [Release from Grid](#). It is also possible to select a distribution function for the initial mass. This is important when modeling separation devices where the goal is to understand the *transmission probability* of particles of various sizes.

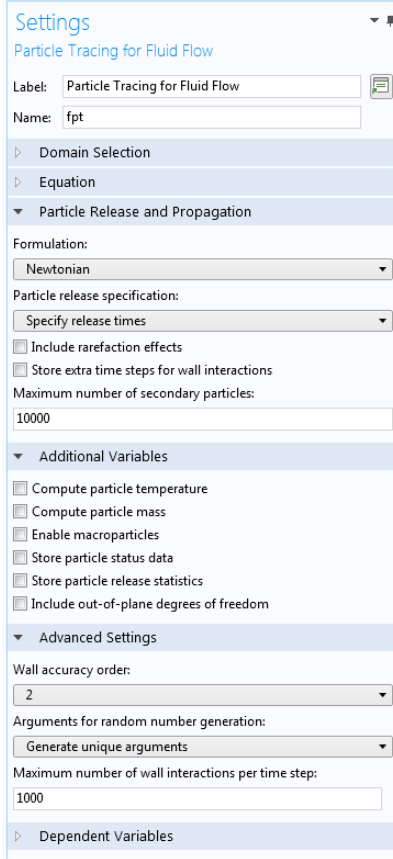


Figure 2-3: Settings window for the Particle Tracing for Fluid Flow interface.

## MODELING ADVECTION AND DIFFUSION

Continuum methods such as FEM have one major drawback when it comes to modeling the advection and diffusion of a particulates in a fluid. The higher the *Péclet number*, the more numerically unstable the method becomes. The Péclet number is the ratio of the rate of advection to the rate of diffusion:

$$P_e = \frac{L|\mathbf{u}|}{D}$$

where  $L$  is the characteristic dimension,  $\mathbf{u}$  is the advection velocity, and  $D$  is the diffusion coefficient of the particles. In general, continuum methods cannot handle

systems where the Péclet number is greater than around 1000. The diffusion coefficient  $D$  (SI unit:  $\text{m}^2/\text{s}$ ) for spherical particles is, following Einstein's relation,

$$D = \frac{k_B T}{6\pi\mu r_p}$$

where

- $\mu$  (SI unit:  $\text{Pa}\cdot\text{s}$ ) is the fluid dynamic viscosity,
- $k_B = 1.3806488 \times 10^{-23}$  J/K is Boltzmann's constant,
- $T$  (SI unit: K) is the temperature, and
- $r_p$  (SI unit: m) is the radius of the particles.

For 100 nm diameter particles in water at room temperature, this results in a diffusion coefficient of around  $4 \times 10^{-12}$   $\text{m}^2/\text{s}$ . For a microfluidic device with characteristic size 1 mm and velocity of 1 m/s, this results in a Péclet number of  $2.5 \times 10^8$ . Handling such a large Péclet number with continuum methods is clearly not possible (Ref. 1).

Particle trajectories are computed in a Lagrangian reference frame, removing the restriction on the Péclet number. The Péclet number can be anything from 0 to infinity without introducing numerical instabilities. Advection is added to the particles via the [Drag Force](#). Diffusion is added to particles by adding the [Brownian Force](#). If the background velocity field is zero then particle motion is purely diffusive (zero Péclet number). If the Brownian force is neglected and the background velocity is nonzero, the motion is pure advection (infinite Péclet number).

### PARTICLES IN A RAREFIED GAS

Many drag laws, such as the Stokes drag law, are based on the assumption of continuum flow, in which the particle Knudsen number  $\text{Kn}$  (dimensionless) is very small,

$$\text{Kn} = \frac{\lambda}{L} \ll 1$$

where  $\lambda$  (SI unit: m) is the mean free path of molecules in the surrounding fluid, and  $L$  (SI unit: m) is a characteristic length of the particle, which is often the particle radius or diameter. The exact definition of the characteristic length may vary depending on the source being cited and should be considered with caution.

When the particles are extremely small or they are surrounded by a rarefied gas, the assumption of continuum flow may not be valid. By selecting the **Include rarefaction**

**effects** check box in the physics interface **Particle Release and Propagation** section, it is possible to apply correction factors to the [Drag Force](#) and [Thermophoretic Force](#), enabling accurate modeling of particle motion in the slip flow, transitional flow, and free molecular flow regimes.

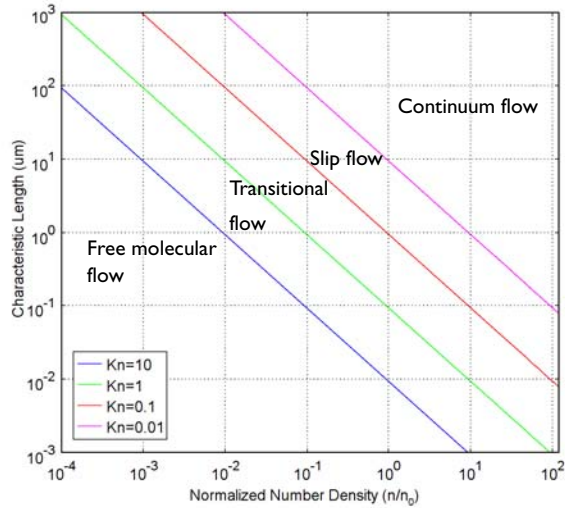


Figure 2-4: A plot showing the main fluid flow regimes for rarefied gas flows. Different regimes are separated by lines of constant Knudsen numbers. The number density of the gas is normalized to the number density of an ideal gas at a pressure of 1 atmosphere and a temperature of 0° C ( $n_0$ ).

## MODELING MOTION OF LIQUID DROPLETS

Several built-in features are available for modeling the formation, propagation, and breakup of sprays of liquid droplets.

The [Droplet Breakup](#) feature and its two subnodes, [Kelvin-Helmholtz Breakup Model](#) and [Rayleigh-Taylor Breakup Model](#), can be used to model the breakup of liquid droplets into successively smaller child droplets. The [Nozzle](#) feature can be used to release a spray of droplets at a specified location.

The [Droplet Sprays in Fluid Flow Interface](#) functions like [The Particle Tracing for Fluid Flow Interface](#), with appropriate default features and settings for modeling the breakup of liquid droplets.

## *Mathematical Particle Tracing*

---

The [Mathematical Particle Tracing Interface](#) gives access to the underlying mathematical formalism on which the Charged Particle Tracing and Particle Tracing for Fluid Flow interfaces are built. The Mathematical Particle Tracing interface allows for specification of particle motion in terms of either a *Lagrangian* or *Hamiltonian*. Often it is easier to write down an expression for the Lagrangian or Hamiltonian for particles rather than deriving the equations of motion. The Hamiltonian formulation solves for both the particle position and the particle momentum, so the number of degrees of freedom is doubled when the Hamiltonian formulation is activated.

## *Multiphysics Interfaces*

---

The Charged Particle Tracing and Particle Tracing for Fluid Flow interfaces can be used to create unidirectional, or one-way, couplings between particles and fields. For example, when using the Charged Particle Tracing interface, it is possible to compute the electric potential using a Stationary study and then use the computed potential to exert an electric force on the particles.

It is also possible to create bidirectional couplings, in which the particles contribute to fields in the surrounding domains, which may then exert forces on the particles. A typical example is a diverging beam of charged particles.

Dedicated Multiphysics nodes are available for several of the most common examples of bidirectional coupling between particles and fields. The following Multiphysics nodes are available depending on the other physics interfaces that are present:

- [Electric Particle Field Interaction](#)
- [Space Charge Limited Emission](#)
- [Magnetic Particle Field Interaction](#)
- [Fluid-Particle Interaction](#)

In addition, it is possible to add all of the physics interfaces necessary for a specific Multiphysics node by using the dedicated Multiphysics interfaces:

- [The Particle Field Interaction, Non-Relativistic Interface](#)
- [The Particle Field Interaction, Relativistic Interface](#)
- [The Fluid-Particle Interaction Interface](#)

For example, to model a non-relativistic beam of electrons that diverges due to self-potential, add the Particle Field Interaction, Non-Relativistic interface. This

automatically adds instances of the Charged Particle Tracing and Electrostatics physics interfaces and the **Electric Particle Field Interaction** Multiphysics node.

# Modeling Tools

In this section:

- [Choosing a Formulation](#)
- [Geometry and Mesh Settings](#)
- [Special Variables](#)
- [Component Couplings](#)
- [Monte Carlo Modeling](#)
- [Study Setup](#)
- [Auxiliary Dependent Variables and Residence Time](#)
- [Initialization of Auxiliary Dependent Variables](#)
- [Accumulators](#)
- [Overriding Particle Properties](#)
- [Filters](#)
- [Particle Counters](#)
- [Improving Plot Quality](#)
- [References](#)



See [Particle Tracing Plots, Data Sets, Derived Values, and Studies](#) for links to the *COMSOL Multiphysics Reference Manual*.

---

## *Choosing a Formulation*

---

The formulation of the equations of motion is specified by selecting an option from the **Formulation** list in the physics interface **Particle Release and Propagation** section.

### **NEWTONIAN**

The **Newtonian** formulation is the default and the most common formulation. It defines a set of second-order ordinary differential equations for the components of the particle position based on Newton's 2nd law of motion,

$$\frac{d}{dt}\left(m_p \frac{d\mathbf{q}}{dt}\right) = \mathbf{F}$$



where  $\mathbf{q}$  (SI unit: m) is the particle position,  $m_p$  (SI unit: kg) is the particle mass, and  $\mathbf{F}$  (SI unit: N) is the total force on the particles.

### **NEWTONIAN, FIRST ORDER**

The **Newtonian, first order** formulation is an alternative to the **Newtonian** formulation. It defines a set of coupled first-order ordinary differential equations for the components of the particle position and velocity.

The default time stepping method for the first-order Newtonian formulation is the **Dormand-Prince 5** Runge-Kutta method, an explicit time stepping method. By comparison, the second-order Newtonian formulation uses the **Generalized alpha** implicit method by default.

The explicit time stepping method is less suitable for stiff problems, meaning that the **Newtonian** formulation is more robust when the particles are subjected to extremely large, abrupt accelerations. The **Newtonian** formulation is also favorable for problems involving ultrarelativistic particles. However, for some nonstiff problems, the explicit method can give comparable or even better accuracy and performance, compared to the implicit method.

Generally, it is most convenient to begin with the default **Newtonian** formulation, then to consider switching to the **Newtonian, first order** formulation to optimize performance if the problem is not overtly stiff.



*Motion of Trapped Protons in Earth's Magnetic Field: Application*  
Library path **Particle\_Tracing\_Module/Charged\_Particle\_Tracing/**  
**trapped\_protons**

---

### **MASSLESS**

The **Massless** formulation defines a set of first-order ordinary differential equations for the components of the particle position only. The particle velocity is directly specified, either by an expression of by using a previously computed field.

The **Massless** formulation is useful for modeling the motion of extremely small particles in a fluid. If Brownian diffusion and turbulent dispersion are neglected, the ratio of the drag force to the mass of an extremely small particle is often so large that the particle reaches equilibrium with the surrounding flow almost instantaneously, so that the particles follow fluid streamlines. Such particles are sometimes referred to as *tracer particles*, as opposed to *inertial particles* which are allowed to have some nonzero velocity relative to the fluid.

## LAGRANGIAN

The **Lagrangian** formulation (Ref. 2, Chapter 1) defines a set of second-order ordinary differential equations for the components of the particle position. Compared to the **Newtonian** formulation, the **Lagrangian** formulation has the same number of degrees of freedom and offers greater flexibility in specifying the equations of motion, but it is not possible to use the **Force** feature.

The equation of motion for a system with Lagrangian  $L$  (SI unit: J) is

$$\frac{d}{dt} \left( \frac{\partial L}{\partial \mathbf{v}} \right) = \frac{\partial L}{\partial \mathbf{q}} \quad (2-1)$$

where  $\mathbf{v}$  (SI unit: m/s) is the particle velocity and  $\mathbf{q}$  (SI unit: m) is the particle position. Equation 2-1 is called the *Euler-Lagrange* equation.

The Lagrangian of a free, nonrelativistic particle is

$$L = T = \frac{1}{2} m_p (\mathbf{v} \cdot \mathbf{v}) \quad (2-2)$$

For example, for an instance of [The Mathematical Particle Tracing Interface](#) with tag `pt` in 3D, the expression `pt.mp*(pt.vx^2+pt.vy^2+pt.vz^2)/2` is the Lagrangian for a free particle that is not subjected to any forces. Note that substitution into the Euler-Lagrange equation yields

$$m_p \frac{d\mathbf{v}}{dt} = \mathbf{0}$$

If all forces can be expressed as the gradients of potentials, it is possible to specify Newton's law of motion in terms of a Lagrangian,

$$L = T - U$$

where  $T$  (SI unit: J) is the particle kinetic energy and  $U$  (SI unit: J) is the total potential energy. For example, if  $U$  only depends on particle position, not velocity, then substitution into the Euler-Lagrange equation yields

$$m_p \frac{d\mathbf{v}}{dt} = -\nabla U$$

## HAMILTONIAN

The Hamiltonian formulation defines a set of coupled first-order ordinary differential equations for the components of the particle position and generalized momentum.

Following Chapter 7 in Ref. 2 the Hamiltonian  $H$  (SI unit: J) can be derived directly from an expression for the Lagrangian  $L$ . The degrees of freedom are the position vector components  $q_i$  and the *generalized momenta*  $p_i$ , defined as

$$p_i = \frac{\partial L}{\partial \dot{q}_i} \quad \dot{q}_i = \frac{dq_i}{dt}$$

For example, for a free, nonrelativistic particle, the generalized momenta are

$$p_i = \frac{\partial}{\partial \dot{q}_i} \left( \frac{1}{2} m_p \sum_j \dot{q}_j^2 \right)$$

Where the sum is over space dimensions in the model. This yields the simplified result

$$p_i = m_p \dot{q}_i \tag{2-3}$$

for  $i$  from 1 to the total number of space dimensions. Thus, in this case the generalized momentum is simply the particle momentum.



For some definitions of the Hamiltonian, the generalized momentum and particle momentum may differ, so Equation 2-3 is not necessarily true for any arbitrary Hamiltonian. However, certain features that accept expressions for momentum components, like the **General reflection** condition for the **Wall** feature, treat the specified expressions as values of the particle momentum, not the generalized momentum. When using the Hamiltonian formulation, always begin by checking whether the Equation 2-3 holds, and use extra caution when entering user-defined expressions for momentum components if it does not.

The Hamiltonian is then defined as

$$H = \sum_j p_j \dot{q}_j - L$$

Using this Hamiltonian the following first-order equations are defined:

$$\frac{dq_i}{dt} = \frac{\partial H}{\partial p_i} \quad \frac{dp_i}{dt} = -\frac{\partial H}{\partial q_i} \tag{2-4}$$

These are known as *Hamilton's equations*.

For example, for a free particle,

$$H = \sum_j p_j \dot{q}_j - \sum_j \frac{p_j^2}{2m_p}$$

Substitution with [Equation 2-2](#) and [Equation 2-3](#) then yields

$$H = \frac{1}{2m_p}(\mathbf{p} \cdot \mathbf{p})$$

For example, for an instance of [The Mathematical Particle Tracing Interface](#) with tag `pt` in 3D, using default names for the generalized momentum components, the expression  $(px^2+py^2+pz^2)/(2*pt.mp)$  is the Hamiltonian for a free particle that is not subjected to any forces. Substitution into Hamilton's Equations then yields

$$\frac{d\mathbf{q}}{dt} = \frac{1}{m_p}\mathbf{p} \quad \frac{d\mathbf{p}}{dt} = 0$$

As expected, the particle moves in a straight line and its momentum is conserved.

### *Geometry and Mesh Settings*

---

When particles reach the boundaries of geometric entities in a model, they do not interact with an exact parameterized representation of the geometry. Rather, they propagate through the mesh elements that discretize the modeling domain and interact with the boundary elements that cover the surfaces of the geometric entities.

#### **REPRESENTATION OF CURVED SURFACES**

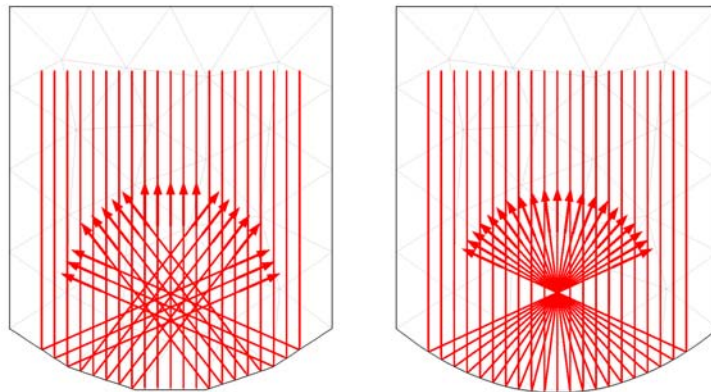
When the surfaces of the geometry are flat, the shape of the surface mesh is indistinguishable from the shape of the geometric entities themselves. Therefore, the fact that particles interact with the mesh instead of the geometry does not introduce any discretization error, and it is possible to accurately compute particle trajectories even when the mesh is extremely coarse.

Curved surfaces in the geometry, however, usually incur a significant amount of discretization error when predicting how particles will interact with them. The time and location at which the particle interacts with the boundary mesh element might be slightly different from the time at which it would have interacted with an exact representation of the surface. In addition, the tangential and normal directions on the boundary mesh element may differ from the tangential and normal directions on the surface, affecting the accuracy of boundary conditions that involve the tangential and

normal directions, such as the **Bounce** condition which causes particles to undergo specular reflection.

The order of the curved mesh elements used to determine the geometry shape is controlled by the **Geometry shape order** list in the **Model Settings** section of the **Settings** window for the main **Component** node. If **Automatic**, the default, is selected, the curved mesh elements are usually represented by quadratic curves; in some cases, linear functions are used to prevent inverted mesh elements from being created.

The effect of the geometry shape order is most notable on a coarse mesh, as shown in [Figure 2-5](#). The mesh elements are shown as pale gray lines in the background and the particle trajectories are represented as thick red arrows. The particles initially move downward and are specularly reflected by a parabolic surface. If **Linear** is selected from the **Geometry shape order** list, all particles that hit the same boundary element are specularly reflected in the same direction, as shown on the left. Even though the bottom surface is parabolic, the particles don't all intersect at a single focus due to the discretization error. If **Quadratic** or **Automatic** is selected, particles that hit the same boundary element can still be reflected in different directions because the tangential and normal directions can vary along the surface of the curved element. As a result, the particles all intersect at a well-defined focal point.



*Figure 2-5: Comparison of particles being specularly reflected at a curved boundary represented using linear elements (left) and quadratic elements (right).*

#### **PARTICLE TRACING IN AN IMPORTED MESH**

It is also possible to compute particle trajectories in an imported mesh. The mesh can be imported from a COMSOL Multiphysics file (.mphbin for a binary file format

or .mpltxt for a text file format) or from a NASTRAN file (.nas, .bdf, .nastran, or .dat).

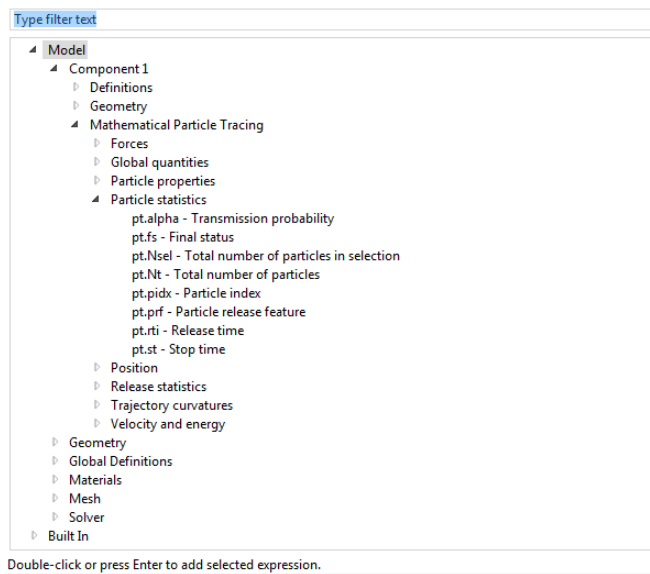
If the mesh is imported from a COMSOL Multiphysics file, the imported mesh always uses linear geometry shape order for the purpose of modeling particle-boundary interactions, even if the model used to generate the mesh had a higher geometry shape order.

If the mesh is imported from a NASTRAN file, the particle-boundary interactions may be modeled using either linear or higher geometry shape order. If **Export as linear elements** is selected when generating the NASTRAN file, or if **Import as linear elements** is selected when importing the file, then linear geometry shape order will be used.

### *Special Variables*

---

The particle tracing interfaces define a number of special variables, some of which can only be used during results processing. These variables can be found in the **Particle statistics** plot group during results processing, as shown in [Figure 2-6](#). All of the variables described in this section are preceded by its Name; see [The Mathematical Particle Tracing Interface](#) for more information.



*Figure 2-6: An example of variables available from the particle statistics menu and available with the Mathematical Particle Tracing interface.*

The following variables are defined:

- **Particle index**, `pidx`. Each particle is assigned a unique index starting from 1 up to the total number of particles. This expression can be passed into a function, which can create, for example, random forces that are unique for each particle. 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)`.
- **Particle release feature**, `prf`. If there are multiple release features in a model, it is useful to be able to visualize how the particles mix together based on their initial release position. The **Particle release feature** variable takes a numeric value, starting at 1, which is unique to each release feature.



When you also have the CFD Module, see *Particle Tracing in a Micromixer*: Application Library path **CFD\_Module/Particle\_Tracing/micromixer\_particle\_tracing**.

---

- **Particle release time**, `prt`. Since particles can be released at arbitrary points in time, it is often useful to visualize at what time a specific particle was released. The **Particle release time** is only available for primary particles, not those released due to secondary emission.

If the **Store particle status data** check box is selected in the physics interface **Additional Variables** section, then the following additional variables are created:

- The release time of a given particle (variable name `rti`). This works for secondary particles and thus allows for extraction of the time at which a secondary particle was released.
- The time at which a particle stopped at a boundary (variable name `st`).
- The final status of the particle (variable name `fs`). This indicates the status of a particle at a given point in time. When used during postprocessing, the value always

indicates the status of the particle at the last time step. The value is an integer which has one of the following values:

- 0 for unreleased particles.
  - 1 for particles that are still in the modeling domain.
  - 2 for frozen particles.
  - 3 for stuck particles.
  - 4 for particles that have disappeared.
- The status of the particle (variable name `particlestatus`, with no scope). This indicates the current status of each particle. The value is an integer with the same meaning as the final status of the particle. When the **Store particle status data** check box is cleared, the variable `particlestatus` exists while computing particle trajectories, so it can be used in weak expressions defined on the particles, but cannot be used during postprocessing.



---

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 particle.

---

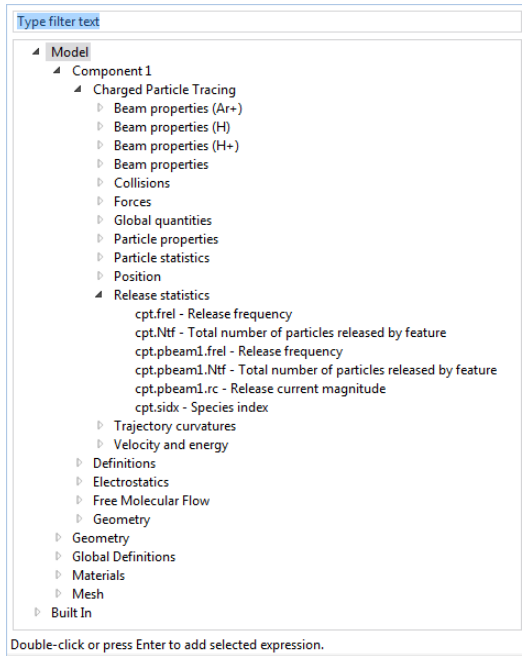
- **Total number of particles**, `Nt`. This total includes both primary and secondary particles, and includes particles that have disappeared or have not been released.
- **Total number of particles in selection**, `Nsel`. If a selection has been applied to the **Particle** data set, the number of particles in that selection can be evaluated.
- **Transmission probability**, `a1pha`. Often the transmission probability is the main quantity of interest in a particle tracing model. The transmission probability is usually computed by dividing the total number of particles which reach an outlet by the number of particles released at an inlet. The **Transmission probability** variable is much more general, and can be used on domains, boundaries, or combinations of both.



- 
- *Particle Trajectories in a Laminar Static Mixer*: Application Library path **Particle\_Tracing\_Module/Fluid\_Flow/laminar\_mixer\_particle**
  - *Brownian Motion*: Application Library path **Particle\_Tracing\_Module/Tutorials/brownian\_motion**
-



The following variables are found in the **Release statistics** plot group during results processing, as shown in [Figure 2-7](#).



*Figure 2-7: An example of variables available from the release statistics menu and available with the Charged Particle Tracing interface.*

- **Total number of particles released by feature**, `<tag>.Ntf`, where `<tag>` is the tag of a particle release feature, such as the [Release](#), [Inlet](#), or [Release from Grid](#) feature. This global variable is uniquely defined for each release feature, and gives the total number of particles that are released by that feature. This includes particles that have not yet been released and particles that have disappeared or become frozen or stuck due to collisions with walls. It does not include any secondary particles.
- **Release frequency**, `<tag>.frel` (SI unit: Hz), where `<tag>` is the tag of a particle release feature. This variable appears when the **Particle release specification** in [The Charged Particle Tracing Interface](#) or [The Particle Tracing for Fluid Flow Interface](#) is set to **Specify current** or **Specify mass flow rate** respectively, and yields the number of physical particles per model particle per second which must be released in order to generate the user defined **Release current magnitude** or **Mass flow rate**, respectively.

- **Release frequency**, `fre1` (SI unit: Hz). This variable also appears when the **Particle release specification** is set to **Specify current** or **Specify mass flow rate**, and is defined for every particle, yielding the release frequency of the feature which produced that particle.
- **Release current magnitude**, `<tag>.rc` (SI unit: A). This variable is available when the **Particle release specification** in [The Charged Particle Tracing Interface](#) is set to **Specify current**, and yields the magnitude of the release current at the release feature identified by `<tag>`.
- **Mass flow rate**, `<tag>.mdot` (SI unit: kg/s). This variable is available when the **Particle release specification** in [The Particle Tracing for Fluid Flow Interface](#) is set to **Specify mass flow rate**, and yields the mass flow rate at the release feature identified by `<tag>`.

### *Component 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 particles, such as the average kinetic energy or the maximum residence time. A particle tracing interface with Name `<name>` creates the following operators:

- `<name>.<name>op1 (expr)` evaluates the sum of the expression `expr` over the particles. The sum includes all particles that are active, frozen, or stuck to boundaries. It excludes particles that have not yet been released and those that have disappeared.
- `<name>.<name>op_all1 (expr)` evaluates the sum of the expression `expr` over all particles, including particles those that are not yet released or have disappeared. Since the coordinates of unreleased and disappeared particles are not-a-number (NaN), the sum may return NaN if the model includes unreleased or disappeared particles. An expression such as `pt.ptop1 (isnan (qx))` can be used to compute the total number of unreleased and disappeared particles.
- `<name>.<name>aveop1 (expr)` evaluates the average of the expression `expr` over the active, frozen, and stuck particles. Unreleased and disappeared particles contribute to neither the numerator nor the denominator of the arithmetic mean.
- `<name>.<name>aveop_all1 (expr)` evaluates the average of the expression `expr` over all particles. It is likely to return NaN if the model includes unreleased or disappeared particles.
- `<name>.<name>maxop1 (expr)` evaluates the maximum value of the expression `expr` over all active, frozen, and stuck particles.

- `<name>.<name>maxop_all1(expr)` evaluates the maximum value of the expression `expr` over all particles.
- `<name>.<name>minop1(expr)` evaluates the minimum value of the expression `expr` over the active, frozen, and stuck particles.
- `<name>.<name>minop_all1(expr)` evaluates the minimum value of the expression `expr` over all particles.
- `<name>.<name>maxop1(expr, evalExpr)` evaluates the expression `evalExpr` for the particle that has the maximum value of the expression `expr` out of all active, frozen, and stuck particles. For example, in a model that uses the Mathematical Particle Tracing interface with name `pt`, the expression `pt.ptmaxop1(pt.V, qx)` would evaluate the  $x$ -coordinate `qx` of the particle with the greatest velocity magnitude `pt.V`.
- `<name>.<name>maxop_all1(expr, evalExpr)` evaluates the expression `evalExpr` for the particle that has the maximum value of the expression `expr` for all particles, including disappeared and unreleased particles.
- `<name>.<name>minop1(expr, evalExpr)` evaluates the expression `evalExpr` for the particle that has the minimum value of the expression `expr` out of all active, frozen, and stuck particles. For example, in a model that uses the Mathematical Particle Tracing interface with name `pt`, the expression `pt.ptminop1(pt.V, qx)` would evaluate the  $x$ -coordinate `qx` of the particle with the smallest velocity magnitude `pt.V`.
- `<name>.<name>minop_all1(expr, evalExpr)` evaluates the expression `evalExpr` for the particle that has the minimum value of the expression `expr` for all particles, including disappeared and unreleased particles.

An instance of the Mathematical Particle Tracing interface with the default name `pt` defines the built-in component couplings shown in [Table 2-1](#).

TABLE 2-1: BUILT-IN COMPONENT COUPLINGS FOR THE MATHEMATICAL PARTICLE TRACING INTERFACE

NAME	DESCRIPTION
<code>pt.pttop1(expr)</code>	Sum over particles
<code>pt.pttop_all1(expr)</code>	Sum over all particles
<code>pt.pttaveop1(expr)</code>	Average over particles
<code>pt.pttaveop_all1(expr)</code>	Average over all particles
<code>pt.ptmaxop1(expr)</code>	Maximum over particles
<code>pt.ptmaxop_all1(expr)</code>	Maximum over all particles
<code>pt.ptminop1(expr)</code>	Minimum over particles

TABLE 2-1: BUILT-IN COMPONENT COUPLINGS FOR THE MATHEMATICAL PARTICLE TRACING INTERFACE

NAME	DESCRIPTION
pt.ptminop_all1(expr)	Minimum over all particles
pt.ptmaxop1(expr, evalExpr)	Evaluate at maximum over particles
pt.ptmaxop_all1(expr, evalExpr)	Evaluate at maximum over all particles
pt.ptmaxop1(expr, evalExpr)	Evaluate at minimum over particles
pt.ptmaxop_all1(expr, evalExpr)	Evaluate at minimum over all particles

### *Monte Carlo Modeling*

Many particle tracing physics features define expressions that include random numbers. In the Particle Tracing for Fluid Flow interface, the [Drag Force](#) and [Brownian Force](#) physics features are forces that potentially include components which are random in nature. In addition, some settings for particle-wall interaction and emission of secondary particles depend on random numbers. When these features are included in a model, it can be necessary to solve the problem multiple times and take a statistical average of the results. The seeding of random numbers is controlled by the **Arguments for random number generation** list in the physics interface **Advanced Settings** section. The following options are available:

- **Generate unique arguments** — seed random numbers based on the position of each physics feature in the Model Tree. This ensures that random numbers generated for different physics features are independent of each other. Use this option to make the results of a study reproducible when running the study multiple times in succession. The random numbers may still differ when running the same study on different architectures or in different versions of the software.
- **Generate random arguments** — seed random numbers using randomly generated double-precision numbers. The seeds are regenerated whenever the study is run, causing the results to differ when running the same study multiple times.
- **User defined** — When this option is selected, additional text fields appear in the settings windows for all features that use random numbers. The number entered in this text field is used as an additional argument for random number generation. A set of distinct solutions can be obtained by running a **Parametric Sweep** over several values of this additional argument.

For more information about the available options, see [Particle Release and Propagation](#) in [The Mathematical Particle Tracing Interface](#).

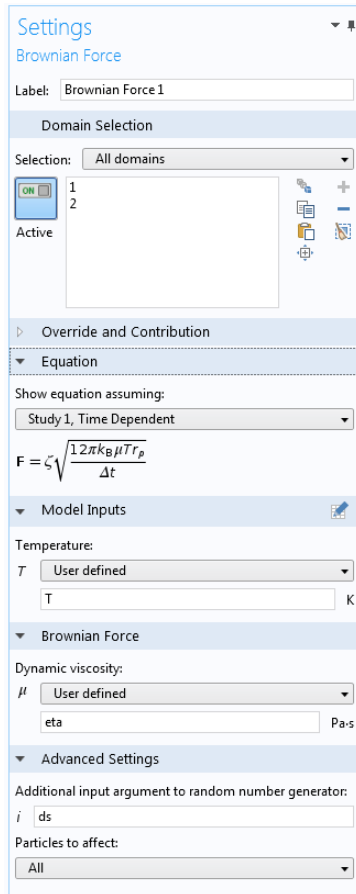


Figure 2-8: An example of the Settings window for Brownian Force.

All three particle tracing interfaces include a [Velocity Reinitialization](#) feature. This allows for general purpose Monte Carlo modeling because the velocity vector can be discretely changed at each time step according to some logical expression. The Charged Particle Tracing interface also includes dedicated features for reinitializing

particle velocity to model the interaction of model particles with molecules in a rarefied background gas.

The [Collisions](#) node supports a variety of subnodes, such as [Elastic](#) collisions and [Ionization](#) reactions, which can be used to model the interactions that can occur when particles collide with atoms or molecules in a rarefied background gas. If a collision should occur, the charged particle has its velocity vector reinitialized by sampling the velocity of the background gas particle at random from a drifting Maxwellian distribution. This provides an accurate description of a charged particle interacting with a background gas.

### *Study Setup*

---

For all of the particle tracing physics interfaces, the particle trajectories must be computed using a Time Dependent solver. For single-physics particle tracing, this can be set up by using the Time Dependent study.

#### **UNIDIRECTIONAL COUPLINGS**

If a unidirectional, or one-way, coupling between particle trajectories and fields is set up, it is often useful to first compute the fields in the surrounding domains and then to compute the particle trajectories using a Time Dependent study. For example, it is possible to model the motion of particles in a channel containing fluid by first setting up the Laminar Flow interface and computing the fluid velocity and pressure using a Stationary study. Then couple the flow profile to the Particle Tracing for Fluid Flow interface using the [Drag Force](#) feature and solve for the particle trajectories in the time domain.

The unidirectional coupling can be used when the particles do not significantly perturb the surrounding fields. In the example of a distribution of particles in a fluid, this means that the force exerted by particles on the fluid is not large enough to noticeably redirect the flow. When modeling the motion of charged particles in an external electric field, this means that the density of charged particles is not large enough to significantly affect the electric potential.

#### **BIDIRECTIONAL COUPLINGS**

If the particles do significantly perturb the fields in the surrounding domains, it is necessary to create a bidirectional, or two-way, coupling. The most straightforward way to set up a bidirectional coupling is by using one of the [Multiphysics Interfaces](#). Alternatively, if the necessary physics interfaces are already present, it is possible to manually add the contributions from the particles to the surrounding fields. For

example, if instances of the Charged Particle Tracing and Electrostatics interfaces are present, the contribution of the charged particles to the space charge density can be included by adding the Electric Particle Field Interaction Multiphysics node.

When setting up bidirectional couplings between physics interfaces, it is important to determine whether the fields are stationary or not. For example, if a beam of charged particles is released at constant current, then it is possible for the space charge density at any point in the beam to remain constant over time.

#### *Bidirectionally Coupled Time Domain Calculation*

If the fields are not constant, it is necessary for the particle trajectories and fields to be computed together in the time domain. For a charged particle beam, this might mean that the beam is pulsed, and thus the electric potential is time-dependent. Full time-domain calculation of the particles and fields is much more computationally demanding than a unidirectional coupling between particles and fields, because it requires particles to be released at a very large number of time steps. For fully time-domain calculations, **Specify release times** should be selected from the **Particle release specification** list in the physics interface **Particle Release and Propagation** section.

#### *Using Iterative Solver Loops*

If the fields are stationary, it is possible to significantly reduce the computational cost of the model by using an iterative solver loop in which the particle trajectories are computed using a Time Dependent solver and the fields are computed using a Stationary solver. To obtain a self-consistent solution, it is necessary to ensure that the result from each of these solvers is used to set the value of variables not solved for by the other solver. The **For** and **End For** nodes, when added to a solver sequence, can be used to set up an iterative loop that does the following:

- 1 Set all contributions from particles to external fields to zero.
- 2 Compute all field variables, using a Stationary solver, using the contributions from particles to external fields computed in the previous step.
- 3 Compute the particle trajectories and their contributions to external fields, using the field variables computed in the previous step.
- 4 Repeat steps 2 and 3 until a specified number of iterations has been reached or another convergence criterion has been satisfied.

The [Bidirectionally Coupled Particle Tracing](#) study step automatically sets up an iterative solver loop, and can be used in place of a Time Dependent study to facilitate the calculation of bidirectionally coupled particle trajectories and fields.



---

If you have the AC/DC Module, see *Electron Beam Divergence Due to Self Potential*, Application Library path **ACDC\_Module/Particle\_Tracing/electron\_beam\_divergence**.

---

When solving for the fields using a Stationary study, it is possible to considerably reduce the number of particles in the simulation by selecting **Specify current** or **Specify mass flow rate** from the **Particle release specification** list in the physics interface **Particle Release and Propagation** section. This causes each model particle to represent a number of real particles per unit time, all moving along the same path, and therefore it is only necessary to release particles during the first time step.

### *Auxiliary Dependent Variables and Residence Time*

---

Auxiliary dependent variables are additional degrees of freedom that can be defined on each particle and used to solve additional first-order ordinary differential equations along each particle trajectory. They can be used to help keep track of things like *residence time*, particle trajectory length, integrated shear rate, and so forth. When an [Auxiliary Dependent Variable](#) feature is added to the physics interface an additional ordinary differential equation is solved for each particle.

To compute the residence time of particles:

- 1 From the **Physics** toolbar, **Global** menu, select **Auxiliary Dependent Variable**.
- 2 In the **Settings** window for **Auxiliary Dependent Variable**, **Auxiliary Dependent Variable** section, enter 1 in the  $R$  field.

For each particle, the differential equation

$$\frac{d}{dt}(\text{rp}) = 1$$

is solved, and so the variable  $\text{rp}$  represents the residence time.

To compute the length of the particle trajectories:

- 3 In the **Auxiliary Dependent Variable** section, from the **Integrate** list, select **Along particle trajectory**.

The following differential equation is now solved for each particle:



$$\frac{d}{ds}(rp) = 1$$

and so the variable `rp` is now the total length of the particle trajectory. The initial values for the auxiliary dependent variables are set in the release features included in the model.

By defining an appropriate source term, it is possible to evaluate the time integral or path integral of any quantity that is known at the particle's position. It is also possible to change the value of an auxiliary dependent variable discontinuously when a particle interacts with a boundary by entering an expression for the new value in the **New Value of Auxiliary Dependent Variables** section of the settings window for the [Wall](#) node. The new value of an auxiliary dependent variable may be defined recursively, for example, to count collisions of a particle with the surrounding walls. In addition, the value of an auxiliary dependent variable can be changed when the **Velocity reinitialization condition** specified in a [Velocity Reinitialization](#) feature is satisfied, enabling the variable to be reinitialized at any location within a domain.

Several built-in features are available for creating auxiliary dependent variables for quantities that are frequently used. For example, the **Compute particle mass** and **Compute particle temperature** check boxes in the settings window for [The Particle Tracing for Fluid Flow Interface](#) create auxiliary dependent variables for the particle mass and temperature, respectively. Several built-in features are available for applying heat sources to the particles when the particle temperature is computed.

The residence time can also be computed in a different way by selecting the **Store particle status data** check box. This creates variables for the particle release time and the particle stop time (the time when a particle freezes or sticks to a boundary, or leaves the modeling domain altogether). The residence time is then simply the difference between the two.

It is also possible to integrate a quantity over time by using the `timeint()` operator. The expression `timeint(t1,t2,expr)` evaluates the integral of the expression `expr` from initial time `t1` to final time `t2`. The accuracy of the integral depends on the total number of time steps taken by the solver. The `timeint()` operator offers a convenient way to evaluate the integrals of many different quantities without recomputing the solution.



[timeint](#) and [timeavg](#) in the *COMSOL Multiphysics Reference Manual*

---

## *Initialization of Auxiliary Dependent Variables*

---

In addition to user-defined variables that can be inserted into a model using the [Auxiliary Dependent Variable](#) feature, many physics features and physics interface settings automatically define built-in auxiliary dependent variables for quantities such as particle temperature, mass, and number of collisions. The particle release features include options to control the way in which these auxiliary dependent variables are initialized when particles are released into the model.

### **CONTROLLING THE ORDER IN WHICH VARIABLES ARE INITIALIZED**

When assigning initial values to particle degrees of freedom, the components of the particle position vector are always assigned first. The initial values of the remaining degrees of freedom, which may include velocity or momentum as well as auxiliary dependent variables, may depend on the initial particle position. In addition, the initial value of any variable may be defined in terms of other variables that are initialized before it.

By default, user-defined auxiliary dependent variables are initialized after all other particle variables. However, in the **Initial Value of Auxiliary Dependent Variables** section of the Settings window for each release feature, the text field for the initial value of each auxiliary dependent variable is accompanied by a check box, **Initialize before particle momentum**. When this check box is selected, the corresponding variable is initialized immediately after the particle position vector components and before any other degrees of freedom. If multiple user-defined auxiliary dependent variables are present, the order in which they are initialized is determined by their relative position in the model tree; variables that appear earlier in the model tree are evaluated before variables that appear below them.

The particle degrees of freedom are then initialized in the following order:

- 1 Position vector components.
- 2 User-defined auxiliary dependent variables for which the **Initialize before particle momentum** check box has been selected. These variables are initialized in the order in which the **Auxiliary Dependent Variable** nodes appear in the model tree.
- 3 Particle mass (Particle Tracing for Fluid Flow interface only).
- 4 Particle temperature (Particle Tracing for Fluid Flow interface only).
- 5 Particle velocity or momentum.
- 6 Other built-in auxiliary dependent variables, including the out-of-plane degrees of freedom in a 2D and 2D axisymmetric geometry. This also includes collision

counters created by the **Collisions** node and the various collision types that can be added to it (Charged Particle Tracing interface only).

- 7 User-defined auxiliary dependent variables for which the **Initialize before particle momentum** check box is cleared.

It is possible to express the initial values of auxiliary dependent variables in terms of other particle degrees of freedom, including position, velocity, and other auxiliary dependent variables. However, particle degrees of freedom can only be defined in terms of other degrees of freedom that are initialized before them, that is, quantities that appear earlier in the above list.

#### **RELEASING DISTRIBUTIONS OF AUXILIARY DEPENDENT VARIABLES**

It is possible to assign a single initial value to an auxiliary dependent variable or to sample multiple values from a distribution. The type of distribution can be selected using the **Distribution function** setting in the **Initial Value of Auxiliary Dependent Variables** section of all release features. Select **None** to enter a single initial value, **Normal** to create a normal distribution function, **Lognormal** to create a log-normal distribution function, **Uniform** to create a uniform distribution function, and **List of values** to enter an a list of numerical values. For the **Normal**, **Lognormal**, and **Uniform** distributions, enter a user-defined **Mean** (default 0) and **Standard deviation** (default 1). It is possible to create a set of uniformly distributed numbers between 0 and 1 by selecting the **Uniform** distribution and entering a **Mean** of 0.5 and a **Standard deviation** of  $\sqrt{1/12}$ .

### *Accumulators*

---

Often the quantity of interest in particle tracing simulations involves the interaction of particles with fields defined on a set of domains or boundaries. Examples of particle-domain interactions include the accumulation of space charge density due to ions and electrons, and the volume force exerted by moving particles on the surrounding fluid. Examples of notable particle-boundary interactions include erosion due to the impact of solid particles on a surface, sputtering of surface molecules due to the impact of ions at a high velocity, and boundary loads generated by a stream of particles impinging on a surface.

An **Accumulator** is a physics feature that allows dependent variables that are defined on domain or boundary mesh elements to be affected by particles that interact with those elements. When an **Accumulator** is added directly to a physics interface, it defines a variable, called an accumulated variable, in each mesh element in a set of domains. The value of the accumulated variable in a mesh element is affected by the presence of

particles within that element. Particles may either affect the accumulated variable directly, by changing the value of the accumulated variable within the mesh element occupied by the particle; or by changing the accumulated variable's time derivative.

When an **Accumulator** is added as a subnode to a [Wall](#), [Outlet](#), or [Axial Symmetry](#) node, it defines an accumulated variable on boundary elements in the selection list of its parent node, defining one degree of freedom per element. These degrees of freedom are incremented as particles hit the boundary. For more information on the **Accumulator** features, see [Accumulator \(Boundary\)](#) and [Accumulator \(Domain\)](#).

The **Accumulator** features are available with all particle tracing interfaces. In addition, the following features are available in [The Charged Particle Tracing Interface](#) to define accumulated variables for more specialized applications: [Etch](#), [Surface Charge Density](#), [Current Density](#), and [Heat Source](#).

The following features in [The Particle Tracing for Fluid Flow Interface](#) define accumulated variables for specific applications: [Erosion](#), [Mass Deposition](#), [Boundary Load](#), and [Mass Flux](#).

The dedicated Multiphysics nodes and interfaces described in the section [Multiphysics Interfaces](#) also use special cases of accumulators on domains to model particle-field and fluid-particle interactions.

Because each particle's location is defined as a point whereas the accumulated variables are defined on mesh elements of finite size, accumulators are inherently mesh-dependent. If the accumulated variable is the density of a physical quantity, it tends to become infinitely large in the limit as the mesh elements become infinitesimally small. To obtain reasonable solutions, the number of particles moving through a domain or interacting with the boundary should typically be at least an order of magnitude larger than the total number of domain elements in a typical cross-section of the domain or the total number of elements on the boundary, respectively.

### *Overriding Particle Properties*

---

It is often useful to compute the trajectories of multiple species of particle simultaneously. By default, all features that release particles, such as the [Inlet](#), [Release from Grid](#), and [Secondary Emission](#) features, create particles with properties defined in the **Particle Properties 1** node. These properties can include particle mass, charge number, diameter, and density. The [Override Properties](#) node can be used to create alternative sets of properties, which can then be assigned to groups of particles by

changing the **Inherit properties** setting for a release feature. For example, the **Override Properties** node can be used to create one **Release from Grid** feature that produces ions and a second **Release from Grid** feature that produces electrons.

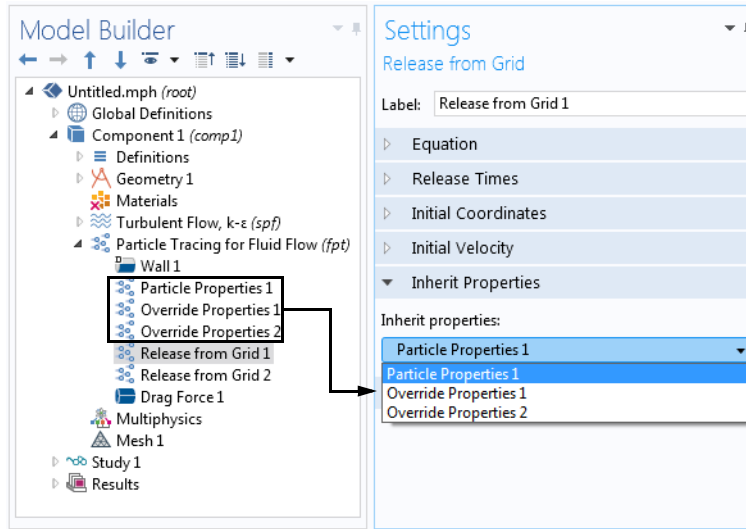


Figure 2-9: The Inherit Properties section of the Settings window for Release from Grid.

## Filters

Visualizing the trajectory of systems with a very large number of particles can consume a lot of computer resources and often particles obscure one another. It is possible to filter the type of particle and the number of particles which should be rendered. To do this, right-click the **Particle Trajectories** plot type and select **Filter**.

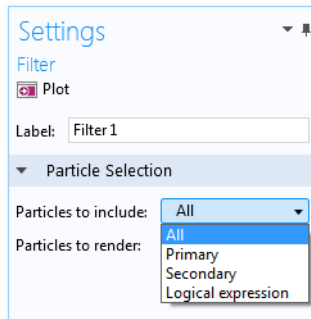


Figure 2-10: An example of the Settings window for the Filter node.

The **Particle type** can be set to render primary particles, secondary particles, both, or by a logical expression.

If particles are obscuring one another or the burden on the graphics card is very high, the number of particles rendered can be reduced by changing the **Particles to render** option. A fraction of the total number of particles to render or the total number of particles to render can be set.

### *Particle Counters*

---

A **Particle Counter** feature is a domain or boundary feature that provides information about particles arriving on a set of selected domains or surfaces from a release feature. Such quantities include the number of particles transmitted, the transmission probability, transmitted current, mass flow rate etc. The feature provides convenient expressions that can be used in the **Filters** node of the **Particle Trajectories** plot, which allows only the particles which reach the particle counter selection to be visualized. The following variables are provided by the **Particle Counter** feature, with the feature tag `<tag>`:

- `<tag>.Nfin` is number of transmitted particles from the release feature to the particle counter at the final time.
- `<tag>.Nsel` is number of transmitted particles from the release feature to the particle counter.
- `<tag>.alpha` is the transmission probability from the release feature to the particle counter.
- `<tag>.rL` is a logical expression for particle inclusion. This can be set in the **Filter** node of the **Particle Trajectories** plot in order to visualize the particles which connect the release feature to the counter.
- `<tag>.It` is the transmitted current from the release feature to the particle counter. This variable is only available for the **Charged Particle Tracing** interface when **Specify current** is selected from the **Particle release specification** list in the physics interface **Particle Release and Propagation** section.
- `<tag>.mdott` is the transmitted mass flow rate from the release feature to the particle counter. This variable is only available for the **Particle Tracing for Fluid Flow** interface **Specify mass flow rate** is selected from the **Particle release specification** list in the physics interface **Particle Release and Propagation** section.

If the release feature is a **Particle Beam** feature in the **Charged Particle Tracing** interface, additional variables for the average position, velocity and energy of the transmitted particles are available.



The **Particle Counter** feature only creates variables, which do not affect the solution. Therefore, they can be added to a model without the need to re-compute the solution, it just needs to be updated. To do this, right click on the **Study** node and select **Update Solution**. The new variables described above will be immediately available for results processing.

### *Improving Plot Quality*

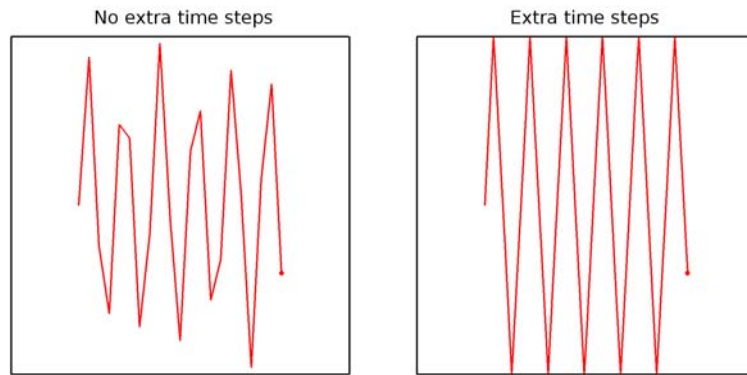
Particles can interact with boundaries at any time during a simulation. However, the solution is only stored at a finite number of time steps. This can sometimes cause the **Particle Trajectories** plot to give misleading results, since the exact instant at which a particle-wall interaction occurs is often not included in the plot.

It is possible to improve the quality of the **Particle Trajectories** plot by selecting the **Store extra time steps for wall interactions** check box in the physics interface **Particle Release and Propagation** section. When this check box is selected, the exact times of particle-wall interactions are stored in memory. When the solution is plotted, a number of extra time steps are included, which are typically close to the times at which particle-wall interactions occur.

In the following example, a particle is released at point (0.2, 0.5) in a unit square with initial velocity (0.6, 11.2). The walls of the square are given the **Bounce** wall condition. The Time-Dependent study is run until 1 s, with output every 0.05 s.

The particle trajectory is computed with the **Store extra time steps for wall interactions** check box cleared and with the check box selected. The results are shown in [Figure 2-11](#).

Note that the solution at the stored times is equally valid in both cases; even though a time step isn't always stored at the exact instant a particle-wall collision occurs, the collision is still handled at the correct time, using an extrapolation algorithm that is determined by the **Wall accuracy order** setting in the physics interface **Advanced Settings** section.



*Figure 2-11: Comparison of the Particle Trajectories plot without extra time steps (left) and with extra time steps (right).*

Selecting the **Store extra time steps for wall interactions** check box does not guarantee that all particle-wall interaction times will be shown in the Particle Trajectories plot. Limitations on the number of extra time steps are imposed to avoid degrading performance by rendering an overwhelming number of particle positions.

For example, consider a model in which 10,000 particles are traced and each particle undergoes approximately 100 wall collisions. There could be as many as 1,000,000 distinct solution times at which a particle-wall collision occurs, each of which would require all 10,000 particles to be rendered because every particle must be shown at the same list of solution times.

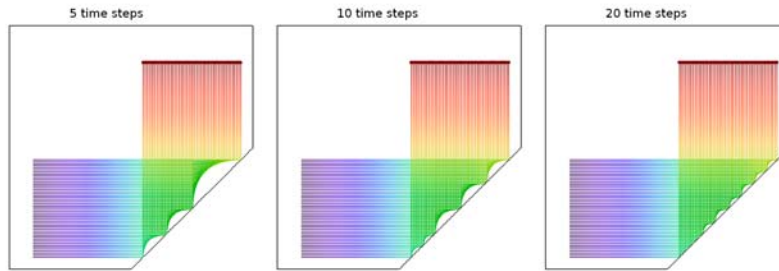
To prevent this, the maximum number of extra time steps shown in the **Particle Trajectories** plot is restricted by options in the plot settings. It is constrained either by an absolute number of extra time steps or by a multiple of the number of time steps stored in the solution.

The following plot illustrates this behavior. In this example, 50 model particles are released with varying initial y-coordinates and initial x-coordinates of 0.2. The particles are specularly reflected by the inclined wall so that they bounce upward.

In the image on the left, a total of 5 time steps are stored in the solution, so no more than 5 additional particle-wall interaction times are plotted. In the middle image, a total of 10 time steps are stored in the solution. In the right image, a total of 20 time steps are stored in the solution. As the number of time steps increases, an increasing number of model particles are shown to touch the wall because increasing the number



of solution times also increases the number of wall interaction times that can be rendered.



*Figure 2-12: Comparison of the stored extra time steps for three different numbers of stored solution times.*

Typically the **Store extra time steps for wall interactions** check box will have a very pronounced effect on models with a small number of model particles, whereas for models with a very large number of particles it may be necessary to specify a very large maximum number of extra steps to generate higher-quality plots.

### *References*

---

1. E.M. Lifshitz and L.P. Pitaevskii, *Fluid Mechanics*, Butterworth, 1987.
2. L.D. Landau and E.M. Lifshitz, *Mechanics*, 3rd ed., Elsevier, 1976.




# Mathematical Particle Tracing

This chapter describes the Mathematical Particle Tracing interface found under the **Mathematics** branch (  $\Delta u$  ) when adding a physics interface.

In this chapter:

- [The Mathematical Particle Tracing Interface](#)
- [Theory for the Mathematical Particle Tracing Interface](#)

# The Mathematical Particle Tracing Interface

The **Mathematical Particle Tracing (pt)** interface (  ), found under the **Mathematics** branch (  $\Delta u$  ) when adding a physics interface, computes the trajectories of particles through a geometry. The particle motion is usually driven by external fields. The particles can be massless or have their motion determined by Newton's second law. You can also specify a Hamiltonian or Lagrangian to dictate the motion of the particles.

When this physics interface is added, these default nodes are also added to the **Model Builder** — **Wall** and **Particle Properties**. Then, from the **Physics** toolbar, add other nodes that implement, for example, boundary conditions and particle release features. You can also right-click **Mathematical Particle Tracing** to select physics features from the context menu.

## SETTINGS

The **Label** is the default physics interface name.

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 `pt`.

## PARTICLE RELEASE AND PROPAGATION

Use the settings in this section to control how particles are released and how the particle trajectories are stored in the solution.

### *Formulation*

Select a **Formulation**— **Newtonian** (the default), **Newtonian, first order**, **Lagrangian**, **Massless**, or **Hamiltonian**. This selection changes the equations of motion that are solved for the particles. It affects the inputs for the [Particle Properties](#) node. For **Newtonian** or **Newtonian, first order** a [Force](#) node can also be added to the model.



For a comparison of the first- and second-order Newtonian formulations, see [Choosing a Formulation](#) in the [Modeling Tools](#) section.

---

#### *Relativistic Correction*

If **Newtonian** or **Newtonian, first order** is selected from the **Formulation** list, the **Relativistic correction** check box is available but by default it is not selected. If you select the check box, then the formulation is applicable for particles with very high speed, meaning you can take relativistic effects on the particle mass into account. The particle mass  $m_p$  (SI unit: kg) is then computed as

$$m_p = \frac{m_r}{\sqrt{1 - \mathbf{v} \cdot \mathbf{v}/c^2}}$$

where  $m_r$  (SI unit: kg) is the rest mass,  $\mathbf{v}$  (SI unit: m/s) is the particle velocity, and  $c = 2.99792458 \times 10^8$  m/s is the speed of light in a vacuum.

#### *Store Extra Time Steps for Wall Interactions*

Typically, the exact times at which particle-wall interactions occur do not coincide with time steps taken by the solver. This does not actually affect the accuracy of the solution, but it can lead to misleading-looking plots when the boundaries reflect particles, because the particle may appear to change direction at a finite distance from the wall, instead of while hitting the wall.

Select the **Store extra time steps for wall interactions** check box to plot the solution at extra time steps, in addition to the specified time steps in the output of the time-dependent solver. Typically these additional time steps are close to times at which the particles hit walls.



For an illustration of the effect of the **Store extra time steps for wall interactions** check box, see [Improving Plot Quality](#) in the [Modeling Tools](#) section.

---

#### *Maximum Number of Secondary Particles*

Some physics features, such as the [Secondary Emission](#) subnode for the [Wall](#) boundary condition, release additional particles other than those that are specified by particle release features. These particles are called *secondary particles* because they are created

as a result of an existing particle interacting with a domain or boundary feature. The memory for all secondary particles must be preallocated when beginning the study.

The **Maximum number of secondary particles** field ensures that sufficient memory for all of the secondary particles is preallocated. It also prevents an inordinate number of particles from being generated by capping them at the number supplied in the field. The default value is 10,000.

If no sources of secondary particles such as [Secondary Emission](#) nodes are present, no unreleased secondary particles are created. If at least one feature is capable of emitting secondary particles, degrees of freedom are allocated for the total number of particles entered in the **Maximum number of secondary particles** text field, even if some of these particles are never released.

### **ADDITIONAL VARIABLES**

Use the settings in this section to determine what additional information is stored in the solution while computing the particle trajectories, such as information about the status of each particle and the entity that released it.

#### *Store Particle Status Data*

By default the **Store particle status data** check box is not selected. When selected it adds new variables for quantities that cannot necessarily be recovered from the particle trajectory data alone. This is especially true if automatic remeshing has been used in a model. The following variables are created:

- The release time of a given particle (variable name `rti`).
- The time at which a particle stopped at a boundary (variable name `st`).
- The final status of the particle (variable name `fs`). This indicates the status of a particle at a given point in time. When used during postprocessing, the value always indicates the status of the particle at the last time step. The value is an integer which has one of the following values:
  - 0 for unreleased particles.
  - 1 for particles which are still in the modeling domain.
  - 2 for frozen particles.
  - 3 for stuck particles.
  - 4 for particles that have disappeared.

In addition, the particle status (variable name `particlestatus`, with no scope) can be used during postprocessing. This variable is always defined while computing the

particle trajectories, but its value is only stored if the **Store particle status data** check box is selected.

To summarize the total number of particles having each final status, the following global variables are also defined.

TABLE 3-1: GLOBAL VARIABLES BASED ON PARTICLE STATUS

NAME	DESCRIPTION
fac	Fraction of active particles at final time step
fds	Fraction of disappeared particles at final time step
ffr	Fraction of frozen particles at final time step
fse	Fraction of secondary particles released
fst	Fraction of stuck particles at final time step

The global variable names in [Table 3-1](#) all take the unreleased secondary particles into account. For example, suppose an instance of the Mathematical Particle Tracing interface includes 100 primary particles and 100 allocated secondary particles. At the last time step, suppose that 80 of the primary particles have stuck to boundaries and that 40 secondary particles have been emitted, all of which are still active. Then the variable `pt.fac`, the fraction of active particles at the final time step, would have the value  $(20 + 40)/(100 + 100)$  or 0.3.

#### *Store Particle Release Statistics*

By default the **Store particle release statistics** check box is not selected. When selected it adds a new internal variable called `releaseindex` to identify the release feature that creates each particle. Every node that is capable of releasing particles, including secondary particles, is associated with a unique positive integer based on its position relative to other nodes in the Model Builder.

When this check box is cleared, the variable `<scope>.prf` can still be used to identify the release feature that creates each particle, but this variable does not account for secondary particle emission.

### **ADVANCED SETTINGS**

This section is only shown when **Advanced Physics Options** are enabled (click the **Show** button on the **Model Builder**).

The **Wall accuracy order** sets the accuracy order of the time stepping used for time steps during which a particle-wall interaction happens:

- Order **1** means that a forward Euler step is used to compute the motion both before and after the wall collision.
- Order **2** (the default) means that a second-order Taylor method is used to compute the motion before the wall collision. After the collision a second-order Runge-Kutta method is used.

The **Wall accuracy order** also controls the time stepping that is used when particles are released during a time step taken by the solver.

Select an option from the **Arguments for random number generation** list: **Generate unique arguments**, **Generate random arguments**, or **User defined**. This setting determines how the additional argument to random functions is defined in features such as the **Wall** boundary condition with the **Diffuse scattering** wall condition and the **Brownian Force** in **The Particle Tracing for Fluid Flow Interface**. Typically the random numbers are functions of the particle index, position, time, and another argument  $i$ , which is defined as follows:

- For **Generate unique arguments** the additional argument is based on the position of each node in the Model Builder. As a result, random numbers generated in different nodes are created independently of each other, but the same result can be reproduced by running the same study several times.
- For **Generate random arguments** the additional argument is randomly created, causing the random functions to return different results each time the study is run.

For **User defined** the additional argument is defined by a user input in the **Settings** window each feature. Independent solutions can be obtained by running a parametric sweep for different values of  $i$ .

Enter a value for the **Maximum number of wall interactions per time step**. The default value is 1000. If a particle undergoes more than the specified number of boundary interactions or velocity reinitializations in a single time step taken by the solver, the particle will disappear. This is included as a safeguard to prevent particles from getting stuck in infinite loops if the time between successive particle-wall interactions becomes infinitesimally small.

## DEPENDENT VARIABLES

The dependent variables (field variables) are the **Particle position**, **Particle position components**, **Particle momentum**, **Particle momentum components**, **Particle velocity**, and **Particle velocity components**. Note that not all of these dependent variables are needed



for every formulation of the equations of motion; for example, the field variable names for **Particle momentum** and **Particle momentum components** are only used if **Hamiltonian** is selected from the **Formulation** list. The names can be changed but the names of fields and dependent variables must be unique within a model.



- [Theory for the Mathematical Particle Tracing Interface](#)
- [Domain, Boundary, Pair, and Global Nodes for the Mathematical Particle Tracing Interface](#)



- *Rössler Attractor*: Application Library path  
**Particle\_Tracing\_Module/Tutorials/rossler\_attractor**
- *Ion Cyclotron Motion*: Application Library path:  
**Particle\_Tracing\_Module/Charged\_Particle\_Tracing/ion\_cyclotron\_motion**

### *Domain, Boundary, Pair, and Global Nodes for the Mathematical Particle Tracing Interface*

The [Mathematical Particle Tracing Interface](#) has these domain, boundary, pair, and global nodes available from the **Physics** ribbon toolbar (Windows users), **Physics** context menu (Mac 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.

#### **DOMAIN**

When **Newtonian** or **Newtonian, first order** is selected as the **Formulation**, the following are available:

- [Force](#)
- [Rotating Frame](#)
- [Particle-Particle Interaction](#)

When **Newtonian**, **Newtonian, first order**, **Lagrangian**, or **Hamiltonian** is selected as the **Formulation**, the following are available:

- Velocity Reinitialization
- Secondary Emission (subnode to **Wall**)
- Secondary Emission (subnode to **Velocity Reinitialization**)

For all **Formulation** choices the following are available:

- Release
- Accumulator (Domain)
- Particle Counter

#### **BOUNDARY**

- Axial Symmetry
- Inlet
- Nonlocal Accumulator (subnode to **Inlet**)
- Outlet
- Wall (default)
- Periodic Condition
- Accumulator (Boundary) (subnode to **Wall**, **Outlet**, and **Axial Symmetry**)
- Particle Counter

#### **PAIRS**

- Inlet
- Particle Continuity

#### **EDGE**

- Release from Edge

#### **POINT**

- Release from Point

#### **GLOBAL**

- Auxiliary Dependent Variable
- Particle Properties (default, singleton)
- Release from Grid
- Override Properties

- [Release from Data File](#)



In the *COMSOL Multiphysics Reference Manual* see [Table 2-3](#) for links to common sections and [Table 2-4](#) 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.

## Wall

Use the **Wall** node to determine what happens to the particles when contact with a wall is made. The [Accumulator \(Boundary\)](#) subnode is available from the context menu (right-click the parent node) or from the **Physics** toolbar, **Attributes** menu. If **Newtonian**, **Newtonian, first order**, **Lagrangian**, or **Hamiltonian** is selected from the **Formulation** list in the physics interface **Particle Release and Propagation** section, the [Secondary Emission](#) subnode is also available.

The **Wall** node can be applied to interior or exterior boundaries, but the instance of this node that is created by default only applies to exterior boundaries.

### WALL CONDITION

Select a **Wall condition** — **Freeze** (the default), **Bounce**, **Stick**, **Disappear**, **Pass through**, **Diffuse scattering**, **Mixed diffuse and specular reflection**, or **General reflection**.

If **Massless** is selected from the **Formulation** list in the physics interface **Particle Release and Propagation** section, only the **Freeze** (default), **Stick**, and **Disappear** conditions are available.

TABLE 3-2: WALL CONDITION OPTIONS

OPTIONS	DESCRIPTION
Freeze	Select to fix the particle position and velocity at the instant a wall is struck. So, the particle position no longer changes after contact with the wall and the particle velocity remains at the same value as when the particle struck the wall. This boundary condition is typically used to recover the velocity or energy distribution of charged particles at the instant contact was made with the wall.
Bounce	Select to specularly reflect from the wall such that the particle kinetic energy is conserved. It is typically used when tracing microscopic particles in a fluid.

TABLE 3-2: WALL CONDITION OPTIONS

OPTIONS	DESCRIPTION
Stick	Select to fix the particle position at the instant the wall is struck. The particle velocity is set to zero. This can be used if the velocity or energy of the particles striking a wall is not of interest.
Disappear	This option means that the particle is not displayed once it has made contact with the wall. Use this option if display of the particle location after contact with the wall is not of interest.
Pass through	When applied to an interior boundary, the Pass through option allows particles to cross the boundary unimpeded. When applied to an exterior boundary, this option behaves like the Disappear option.
Diffuse scattering	Select to bounce particles off a wall according to Knudsen's cosine law. That is, the probability a particle bouncing off the surface in a given direction within a solid angle $d\omega$ is given by $\cos(\theta)d\omega$ , where $\theta$ is the angle between the direction of the reflected particle and the wall normal. The total particle kinetic energy is conserved.
Mixed diffuse and specular reflection	Select to bounce particles off a wall either specularly or according to Knudsen's cosine law, based on a user-defined probability. By combining this wall condition with a primary particle condition, it is possible to include up to three different types of particle-wall interactions at a single boundary. Also enter the Probability of specular reflection $\gamma_s$ .
General reflection	Select to allow an arbitrary velocity to be specified after a particle makes contact with the wall. The velocity components can be functions of the incident particle velocity, energy, or any other quantity. Note that the total momentum of the particle does not necessarily have to be conserved with this option. See <a href="#">General Reflection Settings</a> .

### GENERAL REFLECTION SETTINGS

This section is available when **General reflection** (see [Table 3-2](#)) is selected as the **Wall condition**.

Enter values for the **Reflected particle velocity**  $\mathbf{v}_p$  (SI unit: m/s) either in Cartesian coordinates  $(x, y, z)$  (the default) or select the **Specify tangential and normal velocity components** check box to enter coordinates in the tangent-normal coordinate system  $(t1, t2, n)$ . In this case the normal direction is selected so that an incident particle is reflected back into the domain it previously occupied if the specified normal velocity component is positive. The tangential directions are oriented so that they form a right-handed coordinate system, together with the normal direction. As a result, the tangential and normal directions may point in the opposite direction of the corresponding vectors defined for the geometry (for example, `root.nx`) and for the physics interface (for example, `pt.nx`).

## PRIMARY PARTICLE CONDITION

Select a **Primary particle condition** — **None** (the default), **Probability**, or **Expression**.

When the default, **None**, is kept, it means that the **Wall condition** is always respected by the incident particles.

### *Probability*

If **Probability** is selected, the **Wall condition** is applied with a certain probability. Enter a value for the **Probability**  $\gamma$  (dimensionless) of the particle behaving according to the **Wall condition**. Otherwise the particle behaves according to the [Otherwise](#) setting.



The value of  $\gamma$  should always be between 0 and 1.

---

For example, if the **Wall condition** is set to:

- **Freeze** and  $\gamma$  is set to 0.1, then for every 10 particles that strike the wall, on average one freezes and the remaining 9 particles behave according to the [Otherwise](#) setting.
- **Stick** and  $\gamma$  is set to 0.5, then on average half of the particles stick to the wall and the other half behave according to the [Otherwise](#) setting.

### *Expression*

If **Expression** is selected, the **Evaluation expression**  $e$  (dimensionless) is evaluated whenever the particle strikes the wall. The default value is 1. If the **Evaluation expression** is nonzero, the particle behaves according to the **Wall condition**, otherwise the particle behaves according to the **Otherwise** setting.

### *Otherwise*

The options available for the **Otherwise** setting are the same as for the [Wall Condition](#), except that **Mixed diffuse and specular reflection** and **General reflection** are not available. The **Otherwise** setting can be used to make particles interact with a wall differently when a certain condition is satisfied. For example, to specify that particles with a diameter greater than 1 micron stick to an interior boundary, and the rest pass through:

- Select **Stick** as the **Wall condition**,
- set the **Primary particle condition** to `pt.dp>1E-6`, and
- select **Pass through** as the **Otherwise** option.

## NEW VALUE OF AUXILIARY DEPENDENT VARIABLES

This section is available if an [Auxiliary Dependent Variable](#) has been added to the model.

When a particle crosses or touches a boundary, the values of its auxiliary dependent variables can be changed. The new values can be functions of any combinations of particle variables and variables defined on the boundary. A simple application is to use this to count the number of times a particle strikes the wall.

Select the **Assign new value to auxiliary variable** check box or boxes based on the number of auxiliary variables in the model. Then enter the new value or expression in the field. For example, if there is an auxiliary variable,  $\psi$ , then enter a value for  $\psi_{\text{new}}$  in the field. So, to increment the value of  $\psi$  by 1 when a particle touches or crosses a boundary, enter  $\psi+1$  in the text field for  $\psi_{\text{new}}$ .

## ADVANCED SETTINGS

If the **Primary particle condition** is set to **Probability**, or if the **Diffuse scattering** wall condition is used, then the **Wall** feature generates random numbers. If, in addition, the **Arguments for random number generation** setting is **User defined**, the **Advanced Settings** section is available. Enter the **Additional input argument to random number generator**. The default value is 1.

## *Periodic Condition*

---

The **Periodic Condition** can be used to remove particles that make contact with a source boundary and re-insert them at some other boundary, the destination. The boundaries must be planar and cannot be curved. At least two boundaries must be selected.

## TYPE OF PERIODICITY

Select an option from the **Type of periodicity** list: **Continuity** or **User defined**.

If **Continuity** is selected, the particle velocity at the destination boundary is equal to the velocity at the source boundary.

If the source and destination boundaries are parallel, the components of the particle velocity vector are unchanged. If the source and destination boundaries are not parallel, then the two boundaries are interpreted at the extents of a unit cell exhibiting sector symmetry; thus the velocity vector will be rotated based on the orientation of the destination with respect to the source. This is illustrated in [Figure 3-1](#); the angle between the incident particle velocity  $\mathbf{v}_1$  and the reinitialized particle velocity  $\mathbf{v}_2$  is equal to the angle between the source and destination boundaries (shown in red).

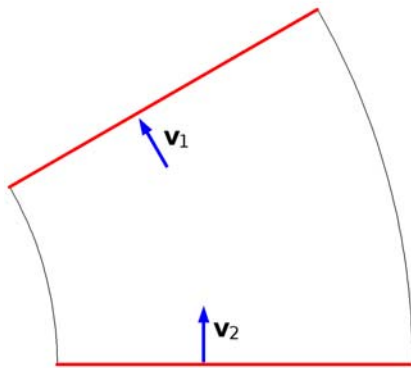


Figure 3-1: Transformation of particle velocity in a geometry exhibiting sector symmetry.

If **User defined** is selected, enter values or expressions for the components of the **Reinitialized particle velocity**  $\mathbf{v}_r$  (SI unit: m/s). The default is zero. Expressions for the reinitialized particle velocity are evaluated at the source boundary, not the destination; for example, expressions in terms of tangential and normal vector components will use the directions tangent or normal to the source boundary. If the velocity is such that the particle will immediately leave the modeling domain through the destination boundary, the particle will instead disappear, to prevent the particle from being subjected to an arbitrarily large number of successive boundary interactions.

#### **NEW VALUE OF AUXILIARY DEPENDENT VARIABLES**

This works in the same way as for the [Wall](#) feature. The expressions are evaluated at the source boundary. See [New Value of Auxiliary Dependent Variables](#).

#### *Accumulator (Boundary)*

---

The **Accumulator** subnode is available from the context menu (right-click the [Wall](#), [Outlet](#), [Periodic Condition](#) or [Axial Symmetry](#) parent 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 particle 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 particle.

## 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 particles that hit the boundary element, and is unaffected by the boundary element size.

Select an option from the **Accumulate over** list: **Particle-wall interactions** (default) or **Particles in boundary elements**.

- For **Particle-wall interactions** the accumulated variable is affected by all particles that hit the boundary element, including those that pass through or are reflected by the boundary.
- For **Particles in boundary elements** the accumulated variable is only affected by particles that freeze or stick to the boundary element.

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 Mathematical Particle Tracing interface using the default variable name `rpb`, the accumulated variable name might be `pt.wall1.bacc1.rpb`.

Enter a **Source  $R$** . The unit of the source term depends on the settings in the **Units** section. Whenever a particle collides 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).

## 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  $\text{mol}/\text{m}^3$ .

## SMOOTHING

The accumulated variables are computed using discontinuous shape functions. Select the **Compute smoothed accumulated variable** check box 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].



### *Particle Counter*

---

Use the **Particle Counter** feature to compute information about particles that are located in a set of selected domains or on a set of selected boundaries. The counter may detect all particles or only the particles released by a specified release feature. Computed variables are the number of particles transmitted, the number of particles transmitted at the final time, the transmission probability, and a logical expression which can be used to filter the rendered particles during results processing.

When used with [The Particle Tracing for Fluid Flow Interface](#), the **Particle Counter** also computes the transmitted mass flow rate when the **Particle release specification** is set to **Specify mass flow rate**.

When used with [The Charged Particle Tracing Interface](#), the **Particle Counter** also computes the transmitted current when the **Particle release specification** is set to **Specify current**. If the release feature is a [Particle Beam](#) feature in the **Charged Particle Tracing** interface, additional variables for the average position, velocity and energy of the transmitted particles are available.

#### **PARTICLE COUNTER**

Select an option from the **Release feature** list. If **All** (the default) is selected, the **Particle Counter** collects information about all particles in the selected domains or boundaries, regardless of how they were released. If any particle release features have been added to the model, they can also be selected from the list. Selecting any particle release feature from the list causes particles from all other sources to be ignored by the counter.

### *Secondary Emission*

---

The **Secondary Emission** subnode is available from the context menu (right-click the [Wall](#), [Axial Symmetry](#), or [Velocity Reinitialization](#) parent node) or from the **Physics**

toolbar, **Attributes** menu. When **Massless** is selected as the **Formulation**, the **Secondary Emission** subnode is not available.



If the **Secondary Emission** node is added to a boundary feature such as a **Wall** or **Axial Symmetry** node, secondary particles are released when particles hit the boundary. If the **Secondary Emission** node is added to a **Velocity Reinitialization** node, secondary particles are released when the particle velocity is reinitialized.

### SECONDARY EMISSION CONDITION

For **Secondary Emission** subnodes applied to boundary nodes, select an option from the **Activate on wall conditions** list: **All**, **Primary only**, or **Otherwise only**. These settings determine which wall conditions in the parent node cause secondary particles to be released.

- For **All** both the **Wall condition** and **Otherwise** condition cause secondary particles to be released.
- For **Primary only** secondary emission only occurs if the incident particle behaves according to the **Wall condition** setting.
- For **Otherwise only** secondary emission only occurs if the incident particle behaves according to the **Otherwise only** setting.

Select a **Secondary emission condition** — **None**, **Probability**, or **Expression**. For **Probability** enter the **Probability**  $\gamma$  (dimensionless) for secondary emission to occur. For **Expression** enter an **Evaluation expression**  $e$  (dimensionless) that must be nonzero for secondary emission to occur.

### SECONDARY PARTICLES

#### *Number of Secondary Particles*

In the **Number of secondary particles** field  $N_s$ , specify the number of secondary particles to release per incident particle. The default value is 1.

#### *Initial Velocity*

Select an **Initial velocity**. If the **Secondary Emission** feature is added to a **Wall** or **Axial Symmetry** node, the available options are **Isotropic hemisphere** (the default), **Reflection of primary particle**, **Diffuse scattering**, and **User defined**. If the **Secondary Emission** feature is added to a domain level feature such as **Velocity Reinitialization**, the available options

are **Constant speed, spherical** and **User defined** (default). The following briefly explains each option.

- **Isotropic hemisphere:** releases the secondary particles with a constant speed and hemispherical velocity direction with the north pole directed in the normal direction away from the wall. The speed of the secondary particles is equal to the speed of the incident particle, divided by the number of secondary particles.
- **Constant speed, spherical:** releases the secondary particles with a constant speed and spherical distribution of velocity directions. Enter a **Speed** with which all secondary particles are released.
- **Diffuse scattering:** causes secondary particles to be released with a probability distribution based on Knudsen's cosine law, like the **Diffuse scattering** option for the [Wall](#) node.
- **Reflection of primary particle:** releases the secondary particles in the direction the primary particle would go if it were specularly reflected, like the **Bounce** option for the [Wall](#) node.
- **User Defined:** allows for an arbitrary velocity vector to be set for the secondary particles. Enter values or expressions for the **Initial particle velocity**  $\mathbf{v}_0$  (SI unit: m/s) either in Cartesian coordinates (x, y, and z for 3D) (the default) or select the **Specify tangential and normal velocity components** check box (available only when the parent node is a **Wall** or **Axial Symmetry** node) to enter coordinates in the tangent-normal coordinate system ( $\tau_1$ ,  $\tau_2$ , n), a specification of the tangential components of the velocity and a normal component (in 2D) and two tangential components and a normal component (in 3D). In this case, the normal is directed away from the wall on which the particle is incident. As a result, the tangential and normal directions may point in the opposite direction of the corresponding vectors defined for the geometry (for example, `root.nx`) and for the physics interface (for example, `pt.nx`).

#### *Scale Velocity by Number of Secondary Particles*

For **Secondary Emission** nodes added to boundary features, select the **Scale velocity by number of secondary particles** check box when the velocity of the secondary particles depends on the velocity of the primary particle to divide the velocity of each secondary particle by the number of secondary particles.

For example, suppose an incident particle with speed 2 m/s hits a **Wall** with a **Secondary Emission** subnode. If the **Secondary Emission** node releases 4 particles in an **Isotropic hemisphere** and the **Scale velocity by number of secondary particles** check box is selected, each reflected particle will have a speed of 0.5 m/s. If the **Scale velocity by number of**

**secondary particles** check box is cleared, each reflected particle will have a speed of 2 m/s.

#### *Offset Initial Position*

This section is only shown if the **Secondary Emission** node is a subnode of a domain-level feature such as the **Velocity Reinitialization** node, not a boundary feature such as **Wall** node.

Select the **Offset initial position** check box to offset the initial positions of the secondary particles before releasing them. The secondary particles can then appear in the region surrounding the parent particle, instead of its exact position.

Select an **Offset method**—**Using initial velocity** (the default) or **Isotropic sphere**.

- For **Using initial velocity** enter a **Time interval**  $\Delta t$  (SI unit: s). The default is 0. Each secondary particle is then displaced by the product of its initial velocity and the specified time interval before being released.
- For **Isotropic sphere** enter a **Particle displacement magnitude**  $\Delta r$  (SI unit: m). The default is 0. All secondary particles are then moved a distance equal to the displacement magnitude before they are released. If the offset would cause secondary particles to be placed outside of the modeling domain, they are instead released at the location of the primary particle.

#### **ADVANCED SETTINGS**

For any of the following combinations of settings, the **Secondary Emission** feature generates random numbers:

- Probability is selected from the **Secondary emission condition** list.
- **Isotropic sphere**, **Diffuse scattering**, or **Constant speed, spherical**, is selected from the **Initial velocity** list.
- The secondary particle positions are offset using the **Isotropic sphere** offset method (shown for domain-level secondary emission only).

If, in addition, **User Defined** is selected from the **Arguments for random number generation** list in the physics interface **Advanced Settings** section, the **Advanced Settings** section is available. Enter the **Additional input argument to random number generator**. The default value is 1.

#### **INHERIT PROPERTIES**

Select an option from the **Inherit Properties** list. The **Particle Properties** node (default) is always available. If any **Override Properties** nodes have been added to the model,

they can also be selected. Use this setting to determine what set of properties is given to the particles that are released by this feature.

### 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 active **Auxiliary Dependent Variable** features in the model, enter a value or expression for its initial value. For example, if the auxiliary dependent variable has the default name `rp`, enter a value or expression for its initial value `rp0`.



### About the Wall Boundary Conditions

---

### *Particle Properties*

---

Use the **Particle Properties** node to set properties based on the **Formulation** selected from the physics interface **Particle Release and Propagation** section. Set the particle velocity for **Massless** formulations, or particle mass for **Newtonian**, **Newtonian, first order**, **Lagrangian**, or **Hamiltonian** formulations.



For theory see:

- [Newtonian Formulation](#)
  - [Hamiltonian Formulation](#)
  - [Lagrangian Formulation](#)
  - [Massless Formulation](#)
- 

### PARTICLE MASS

This section is shown when **Newtonian**, **Newtonian, first order**, **Hamiltonian**, or **Lagrangian** is selected as the **Formulation**.

Enter a value or expression for the **Particle mass**  $m_p$  (SI unit: kg). The default expression is `me_const`, which is a predefined physical constant for the electron mass,  $9.10938188 \times 10^{-31}$  kg.

If the **Relativistic** correction check box is selected in the physics interface **Advanced Settings** section, instead enter a value or expression for the **Particle rest mass**  $m_r$  (SI unit: kg). The default expression is `me_const`.

## LAGRANGIAN

This section is shown for [The Mathematical Particle Tracing Interface](#) and when **Lagrangian** is selected as the **Formulation**.

Enter a value or expression for **Lagrangian**  $L$  (SI unit: J). The default is 0. The Lagrangian is usually a function of the particle kinetic energy and fields, which exert a force on the particles. To include field forces in the Lagrangian, it should be possible to express the field in terms of the gradient of a potential.

## PARTICLE VELOCITY

This section is shown when **Massless** is selected as the **Formulation**.

Enter a vector for the **Particle velocity**  $\mathbf{v}$  (SI unit: m/s) based on space dimension. The **Massless** formulation means that the particles follow streamlines of the particle velocity expression.

## HAMILTONIAN

This section is shown for [The Mathematical Particle Tracing Interface](#) and when **Hamiltonian** is selected as the **Formulation**.

Select an option from the **Specify Hamiltonian** list — **Directly** (the default) or **Manually**.

- For **Directly** enter a value for **Hamiltonian**  $H$  (SI unit: J) The default is 0. The Hamiltonian is typically a function of the particle kinetic energy and any fields which would induce a force on the particles.
- For **Manually** enter coordinates for the **Particle velocity**  $\mathbf{v}_H$  (SI unit: m/s) and **Hamiltonian Force**  $\mathbf{F}_H$  (SI unit: N) based on space dimension.



The **Hamiltonian** formulation solves a system of first order ordinary differential equations for the particle coordinate and the particle momentum. Thus, when using the **Hamiltonian** formulation, the number of degrees of freedom is the same as the **Newtonian, first order** formulation and twice as high as the **Massless, Lagrangian, and Newtonian** formulations.

---



The **Force** node is only available when **Newtonian** or **Newtonian, first order** is selected as the **Formulation** in the physics interface **Particle Release and Propagation** section.

---

Use the **Force** node to exert user-defined forces on particles to influence their motion. All forces defined in the model are added together to compute the total force on the particles. The force can be specified directly or using a susceptibility and field.

### FORCE

Select an option from the **Specify force** list: **Directly** (the default) or **Using susceptibility and field**. For **Directly** enter values or expressions for the **Force  $\mathbf{F}$**  (SI unit: N) based on space dimension.

For **Using susceptibility and field**:

- Select **Isotropic**, **Diagonal**, **Symmetric**, or **Anisotropic** and enter a value or expression in the field or matrix (based on space dimension) for the **Susceptibility  $\chi$** .
- Enter values or expressions for the **Field  $\Gamma$** .

The units of the susceptibility and field are controlled by the [Units](#) section.

### AFFECTED PARTICLES

Select an option from the **Particles to affect** list: **All** (the default), **Single species**, or **Logical expression**.

- If **All** is selected, the force is applied to every particle in the selected domains.
- For **Single species** select an option from the **Affected particle properties** list. The [Particle Properties](#) node is always available and is the default. In addition, if any [Override Properties](#) nodes have been added to the physics interface, they can be selected. Use this option to apply a force to a specific type of particle.
- For **Logical expression** enter a **Logical expression for inclusion  $e$** . The default is 1. The force is applied to all particles in the selected domains for which the logical expression is nonzero.

### UNITS

This section is available if **Using susceptibility and field** is selected from the **Specify force** list. Select a **Susceptibility 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>. The unit of the **Susceptibility** is changed to the specified unit. The unit of the **Field** is changed so that the product of the susceptibility and field has a unit of force.

### *Rotating Frame*

---

Use the **Rotating Frame** feature to exert fictitious forces on particles when computing their trajectories in a rotating frame of reference, including the centrifugal, Coriolis, and Euler forces.

#### **ROTATING FRAME**

In 2D, enter the coordinates of the **Rotation axis base point**  $\mathbf{r}_{bp}$  (SI unit: m) about which the frame rotates. The default is the origin (0,0).

In 3D, select an option from the **Axis of rotation** list: **x-axis**, **y-axis**, **z-axis** (the default), or **User defined**. For **User defined**, enter the coordinates of the **Rotation axis base point**  $\mathbf{r}_{bp}$  (SI unit: m) about which the frame rotates. The default is the origin (0,0,0). Then enter the components of the **Rotation axis direction**  $\mathbf{e}_{ax}$  (dimensionless). The default direction is parallel to the positive  $z$ -axis, (0,0,1). It is not necessary to enter a unit vector for the axis direction because the expression is automatically normalized.

The remaining inputs are the same for any space dimension.

Select an option from the **Rotational direction** list: **Counterclockwise** (the default) or **Clockwise**. The sense of rotation follows the standard right-hand rule; that is, if you curl the fingers on your right hand in the direction of the rotation, your thumb points in the direction of the axis of rotation (in 3D) or in the positive  $z$ -direction (in 2D).

Select an option from the **Rotational frequency** list: **Angular velocity** (the default) or **Revolutions per time**. For **Angular velocity** enter the **Angular velocity magnitude**  $\Omega$  (SI unit: rad/s). The default is 0 [rad/s]. For **Revolutions per time** enter the **Revolutions per time**  $f$  (SI unit: Hz). The default is 0 [rpm].

By default the **Centrifugal force** check box is selected. This means that the fictitious centrifugal force is exerted on the particles.

By default the **Coriolis force** check box is selected. This means that the fictitious Coriolis force, which arises when the particles have nonzero velocity in the rotating frame, is exerted on the particles.



By default the **Euler force** check box is selected. This means that the fictitious Euler force, which arises when the frame has nonzero angular acceleration, is exerted on the particles.

#### AFFECTED PARTICLES

Use this section to exert the fictitious forces on specific particles. The available settings are the same as for the [Force](#) node.



#### Particle Tracing in Rotating Frames

---

#### *Velocity Reinitialization*

---



The **Velocity Reinitialization** node is only available when **Newtonian**, **Newtonian, first order**, **Lagrangian**, or **Hamiltonian** is selected as the **Formulation** in the physics interface **Particle Release and Propagation** section.

---

Use the **Velocity Reinitialization** node to reset the particle velocity to an arbitrary expression if a specific condition is satisfied.

The [Secondary Emission](#) subnode is available from the context menu (right-click the parent node) or from the **Physics** toolbar, **Attributes** menu.

#### VELOCITY REINITIALIZATION

Enter an expression for the **Velocity reinitialization condition**  $e$  (dimensionless). If this expression is nonzero then the effect of the Velocity Reinitialization node is applied to the particle. This expression is evaluated at every time step taken by the solver.

Select the **Specify reinitialization time** check box to make the velocity reinitialization occur at a specific time. If this check box is selected, enter a value or expression for the **Reinitialization time**  $t_r$  (SI unit: s). The default value is 0.



---

If the **Specify reinitialization time** check box is cleared, the reinitialization condition is only evaluated at the beginning of each time step taken by the solver, so each particle's velocity can only be reinitialized once per time step.

If the **Specify reinitialization time** check box is selected, the reinitialization condition and time are reevaluated after every velocity reinitialization and wall interaction, so it is possible for each particle's velocity to be reinitialized multiple times in a single time step taken by the solver.

---

Select an option from the Effect on primary particle list—**None**, **Reinitialize** (the default), **Freeze**, **Stick**, or **Disappear**. If **None** is selected, the trajectory of the primary particle is not affected, although it is still possible to release secondary particles at the primary particle's position and to reinitialize its auxiliary dependent variables.

For **Reinitialize** the velocity is reinitialized at each time step according to the expression specified in the **Reinitialized particle velocity  $\mathbf{v}_r$**  (SI unit: m/s) field. Both the **Velocity reinitialization condition** and the **Reinitialized particle velocity** can be functions of any of the particle variables, that is, particle position, energy, and so forth.

The options **Freeze**, **Stick**, and **Disappear** are identical to the same options described for the [Wall](#) node.

#### **NEW VALUE OF AUXILIARY DEPENDENT VARIABLES**

This section is available if an [Auxiliary Dependent Variable](#) has been added to the model. The values of auxiliary variables can be changed whenever the **Velocity reinitialization condition** is satisfied. See [Wall](#) for all settings.

#### *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 particles in each domain element.

## 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 particles 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 particle 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 particles in that element. If the **Accumulator type** is set to **Density**, this sum is divided by the mesh element volume. As each particle moves through a series of mesh elements, it leaves behind a contribution to the accumulated variable that remains even after the particle has moved on.

Enter the **Accumulated variable name**. The default 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 Mathematical Particle Tracing interface with default name `pt` and default accumulated variable name `rpd`, the variable would be named `pt.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 particles pass through during each time step.

## 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>.



Accumulator Theory: Domains

---

## *Particle-Particle Interaction*

---



The **Particle-Particle Interaction** node is only available when **Newtonian** or **Newtonian, first order** is selected as the **Formulation** in the physics interface **Particle Release and Propagation** section.

This feature does not support hard sphere collisions; the interaction forces must be finite and must be expressed as continuous functions of the distance between particles.

---

Use the **Particle-Particle Interaction** node to define how the particles interact with each other. There are predefined options available for the Coulomb, Lennard-Jones, and linear elastic forces. It is also possible to define arbitrary expressions for the interaction force.

## FORCE

Select an option from the **Interaction force** list: **Coulomb** (the default), **Linear elastic**, **Lennard-Jones**, or **User defined**. If the default, **Coulomb**, is kept, a Coulomb force describes the interaction between charged particles. No user input is necessary.

### *Linear elastic*

For **Linear elastic** enter the **Spring constant**  $k_s$  (SI unit: N/m). The default is 1 N/m. Enter the **Equilibrium distance between particles**  $r_0$  (SI unit: m). The default is 1 mm.

### Lennard-Jones

For **Lennard-Jones** it uses the Lennard-Jones potential to approximate the interaction between neutral particles. The value of these parameters depend on the gas molecules interacting and can usually be found from a literature search.

- Enter the **Collision diameter**  $\sigma$  (SI unit: m), typically in the order of a few angstroms (the default value is  $3.3 \times 10^{-10}$  m or 3.3 Å).
- Enter the **Interaction strength**  $\epsilon$  (SI unit: J), usually in the order of  $10^{-21}$  Joules (the default value is  $1.6 \times 10^{-21}$  J).

### User Defined



There is specific syntax that you must use to specify user-defined interaction forces.

For **User defined** enter a user-defined expression for the interaction **Force**  $\mathbf{F}_u$  (SI unit: N) based on space dimension.

The particle degrees of freedom are given the variable names  $qx$ ,  $qy$ , and  $qz$  (in 3D), but to access the position vector of neighboring particles use the expression  $\text{dest}(qx)$ ,  $\text{dest}(qy)$ , and  $\text{dest}(qz)$ .

A predefined expression for the distance between particles is available because the particle-particle interaction forces usually depend inversely on the distance between the particles. This expression is accessed using  $\langle \text{name} \rangle . r$ , where  $\langle \text{name} \rangle$  is the physics interface node name. So, if the name is  $pt$ , then this variable is accessed using  $pt . r$ .

As an example, the gravitational force on particle  $i$  depends on the position vector and mass of all other particles:

$$\mathbf{F}_i = -Gm^2 \sum_{j=1}^N \frac{(\mathbf{r}_i - \mathbf{r}_j)}{|\mathbf{r}_i - \mathbf{r}_j|^3}$$

where  $\mathbf{r}_i$  is the position vector of the  $i^{\text{th}}$  particle,  $G$  is the gravitational constant, and  $m$  is the mass. To enter this as a user-defined force, enter (in 2D):

```
G*m^2*(qx-dest(qx))/sqrt((qx-dest(qx))^2+(qy-dest(qy))^2+tol)^3  
G*m^2*(qy-dest(qy))/sqrt((qx-dest(qx))^2+(qy-dest(qy))^2+tol)^3
```

where `tol` is a user-defined parameter to prevent divide by zero for the  $i^{\text{th}}$  particle. In practice it is quite difficult to choose the value of `tol`. It should in general be a small fraction of the smallest distance you want to allow between particles.

### USING THE `DEST()` OPERATOR

When reading or writing the expressions for a user-defined particle-particle interaction force, the `dest()` operator is used to identify properties of the particle that is being subjected to a force, while the omission of the `dest()` operator indicates some property or a particle that is exerting the force. For example, when computing the force on the  $i^{\text{th}}$  particle, the expression `qx-dest(qx)` is the difference between the  $x$ -coordinate of some other particle (say, the  $j^{\text{th}}$  particle) and the  $x$ -coordinate of the  $i^{\text{th}}$  particle.

### ADVANCED SETTINGS

The fact that all particles can interact with all other particles in the system means that a full Jacobian matrix is generated at each time step when solving. Assembly and factorization of such a matrix is very expensive in both time and memory.

By default, the **Exclude Jacobian contribution for particle-particle interaction force** check box is selected. This means the contribution to the Jacobian matrix is ignored due to the particle-particle interaction force. This also means that the problem solves much faster and requires much less memory. The drawback of this is that the Jacobian is not exact, and the solver therefore needs to take very small time steps when solving.



*Rotating Galaxy*: Application Library path  
**Particle\_Tracing\_Module/Tutorials/rotating\_galaxy**

---

By default, the **Apply cutoff length** check box is cleared. If this check box is selected, enter a value or expression for the **Cutoff length**  $r_c$  (SI unit: m). The default is 5 mm. If the distance between particles is greater than the cutoff length, the particle-particle interaction force between them is set to zero.


### *Release*

---

Use the **Release** node to release particles into the model from a selected set of domains. The release times, initial position, initial velocity, and initial value of any auxiliary dependent variables can be specified. The sections available are based on the **Formulation** in the physics interface **Particle Release and Propagation** section. If

**Newtonian**, **Newtonian, first order**, **Lagrangian**, or **Hamiltonian** are selected, also enter an initial velocity.

### RELEASE TIMES

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



For models using [The Charged Particle Tracing Interface](#) the **Release Current Magnitude** section is available instead of **Release Times** when the **Particle release specification** is set to **Specify current**. Enter a value for the **Release current magnitude**  $I$  (SI unit: A). The default is 1 nA. For models using [The Particle Tracing for Fluid Flow Interface](#) the **Mass Flow Rate** section is available instead of **Release Times** when the **Particle release specification** is set to **Specify mass flow rate**. Enter a value for the **Mass flow rate**  $\dot{m}$  (SI unit kg/s). The default is 1 mg/s. In either case, all particles are released at time  $t = 0$ . The expression for the **Release current magnitude** or **Mass flow rate** should only depend on parameters and global variables, not on particle variables such as the particle mass.

### INITIAL POSITION

Select an **Initial position**—**Mesh based** (the default), **Density**, or **Random**.

#### *Mesh Based*

For **Mesh based** the particles are released from a set of positions determined by a selection of geometric entities (of arbitrary dimension) in the mesh.

Select a **Refinement factor** between 1 and 5 (the default is 1), where the centers of the refined mesh elements are used. Thus, the number of positions per mesh element is  $\text{refine}^{\text{dim}}$ , except for pyramids, where it is  $(4 * \text{refine}^2 - 1) * \text{refine} / 3$ .

#### *Density*

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

The field **Density proportional to** is an expression—the resulting particle distribution approximately has a density that is proportional to this expression. The resulting

distribution looks a bit random, and it depends on the order in which the mesh elements are numbered.



The distribution is not necessarily the same in different COMSOL versions, but the total number of particles released is always  $N$ .

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 particles to release within each mesh element. The higher the accuracy order, the more accurately particles 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 particle 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 particles within each mesh element is weighted according to the density, but it is possible for some particles to occupy the same initial position. Further increasing the **Position refinement factor** increases the number of evaluation points within each mesh element to reduce the probability of multiple particles occupying the same initial position.



If the model includes a [Particle-Particle Interaction](#) then there is a risk that the force may become infinite if the initial particle positions are not unique. When modeling particle-particle interactions the **Position refinement factor** should be 0 to ensure that initial positions are unique. However, it may then be necessary to refine the mesh to get an accurate initial distribution of particle positions if the density is not uniform.

### *Random*

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

The particle positions are determined in the following way. First a random mesh element is selected for each particle with probability proportional to the element size, so that particles 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. The particle is then placed at this location.



The initial particle coordinates are pseudorandom; that is, they are not derived from a natural entropy source. Therefore, if the same study is re-run on the same architecture multiple times, the same initial particle positions will be used each time.

The **Number of particles per release** is the number before velocity, release time and auxiliary dependent variable multiplication, so the actual number of model particles generated by the **Release** feature may be significantly larger.

### INITIAL VELOCITY

For [The Mathematical Particle Tracing Interface](#) this section is shown when **Newtonian**, **Newtonian, first order**, **Lagrangian**, or **Hamiltonian** is selected as the **Formulation**.

For [The Charged Particle Tracing Interface](#) and [The Particle Tracing for Fluid Flow Interface](#) this section is shown when **Newtonian** or **Newtonian, first order** is selected as the **Formulation**.

Select an option from the **Initial velocity** list: **Expression** (the default), **Kinetic energy and direction**, **Constant speed, spherical**, **Constant speed, hemispherical**, **Constant speed, cone**, **Constant speed, Lambertian** (3D only), or **Maxwellian**.

- For **Expression** enter coordinates for the **Initial particle velocity**  $\mathbf{v}_0$  (SI unit: m/s) based on space dimension. The defaults are 0 m/s.
- For **Kinetic energy and direction** enter the **Initial kinetic energy**  $E_0$  (SI unit: J). The default is 1 keV. Then enter coordinates for the **Initial particle direction**  $\mathbf{L}_0$  (dimensionless) based on space dimension. It is not necessary to normalize the components of the initial direction vector because this is done automatically when computing the initial velocity.



The option **Kinetic energy and direction** is not supported for [The Particle Tracing for Fluid Flow Interface](#).

- For **Constant speed, spherical** a total of  $N_{\text{vel}}$  particles are released at each release point, each with the same speed. The velocity distribution is spherical for 3D components and circular for 2D. Enter the following:
  - **Speed**  $v$  (SI unit: m/s). The default is 1 m/s.
  - **Number of particles in velocity space**  $N_{\text{vel}}$  (dimensionless). The default is 200.
- For **Constant speed, hemispherical** a total of  $N_{\text{vel}}$  particles are released at each release point, each with the same speed. The velocity distribution is hemispherical for 3D components and a half circle for 2D. Enter the:
  - **Speed**  $v$  (SI unit: m/s). The default is 1 m/s.
  - **Number of particles in velocity space**  $N_{\text{vel}}$  (dimensionless). The default is 200.
  - A direction vector for the **Hemisphere axis**  $\mathbf{r}$  (dimensionless). This sets the direction of the north pole of the hemisphere or half circle.
- For **Constant speed, cone** a total of  $N_{\text{vel}}$  particles are released at each release point, each with the same speed. The velocity distribution is a cone of angle  $\alpha$  for 3D components and an arc of half-angle  $\alpha$  for 2D. Enter the:
  - **Speed**  $v$  (SI unit: m/s). The default is 1 m/s.
  - **Number of particles in velocity space**  $N_{\text{vel}}$  (dimensionless). The default is 200.
  - A direction vector for the **Cone axis**  $\mathbf{r}$  (dimensionless). This sets the direction of the cone axis for 3D components or the centerline of the arc for 2D.
  - A **Cone angle**  $\alpha$  (SI unit: rad).
- The **Constant speed, Lambertian** option is only available in 3D. It generates  $N_{\text{vel}}$  particles for each release point, each with the same speed. The velocity distribution is a hemisphere in which the distribution of polar angles follows the cosine law. Enter the:
  - **Speed**  $v$  (SI unit: m/s). The default is 1 m/s.
  - **Number of particles in velocity space**  $N_{\text{vel}}$  (dimensionless). The default is 200.
  - A direction vector for the **Hemisphere axis**  $\mathbf{r}$  (dimensionless). This sets the direction of the north pole of the hemisphere.

- For **Maxwellian** enter the:
  - **Number of particles per velocity space direction**  $N_v$ . The default is 50.
  - **Temperature**  $T_0$  (SI unit: K). The default is 293.15 K.



This option can generate an overwhelming number of particles because for each release time and release position, this number is multiplied by  $N_v^{sdim}$ , where  $sdim$  is the number of space dimensions. Typically only choose this option when releasing particles at a single release time at a single point in space.

For **Constant speed, spherical, Constant speed, hemispherical, Constant speed, cone, Constant speed, Lambertian,** and **Maxwellian**, select an option from the **Sampling from distribution** list: **Deterministic** (the default) or **Random**. If **Deterministic** is selected, the initial velocity is computed by sampling from the velocity distribution in a deterministic and reproducible manner. If **Random** is selected, the particle velocity is sampled from the distribution using pseudo-random numbers and may not always generate the same results.

For **Constant speed, spherical, Constant speed, hemispherical, Constant speed, cone,** and **Constant speed, Lambertian**, select the **Allow nonuniform speeds in distribution** check box to allow particles from the same release point to have different initial speeds. This check box has no effect on the initial velocity directions of the released particles, only their velocity magnitudes. If this check box is selected, expressions for the initial speed in terms of the particle index can return a unique value for each particle in the distribution.



For example, if **Constant speed, hemispherical** is selected from the **Initial velocity** list, the **Allow nonuniform speeds in distribution** check box is selected, and the expression for the initial speed is  $2[\text{m/s}] * (\text{random}(\text{pt.pid}x) + 0.5)$ , then particles will be released with directions sampled from an isotropic hemisphere and with speeds sampled pseudorandomly between 0 and 2 m/s.

## INHERIT PROPERTIES

Select an option from the **Inherit Properties** list. The **Particle Properties** node (default) is always available. If any **Override Properties** nodes have been added to the model,

they can also be selected. Use this setting to determine what set of properties is given to the particles that are released by this feature.

### INITIAL PARTICLE TEMPERATURE

This section is only available for [The Particle Tracing for Fluid Flow Interface](#) and is only shown when the **Compute particle temperature** check box is selected in the physics interface **Additional Variables** section. Enter a value or expression for the **Initial particle temperature**  $T_{p,0}$  (SI unit: K). The default value is 293.15[K].

### INITIAL PARTICLE MASS

This section is only available for [The Particle Tracing for Fluid Flow Interface](#) and is only shown when the **Compute particle mass** check box is selected in the physics interface **Additional Variables** section.

The initial particle mass can be a single value or can be sampled from a distribution. Select an option from the **Distribution function** list—**None** (the default), **Normal**, **Lognormal**, **Uniform**, or **List of values**.

When **None** is selected, enter an initial value  $m_{p,0}$ . The default value is  $10^{-9}$  kg.

Select **Normal** to create a normal distribution function, **Lognormal** to create a log-normal distribution function, or **Uniform** to create a uniform distribution function. The **Number of values**  $N$  (default 1) sets the number of points in the distribution function. Enter the **Mean particle mass** (default  $10^{-9}$  kg) and **Particle mass standard deviation** (default  $10^{-10}$  kg). Select **List of values** to enter a list of numerical values for the initial particle mass directly.

### INITIAL MULTIPLICATION FACTOR

This section is only available for [The Particle Tracing for Fluid Flow Interface](#) and is only shown when the **Enable macroparticles** check box is selected in the physics interface **Additional Variables** section. Enter a value or expression for the **Initial multiplication factor**  $n_{n,0}$  (dimensionless). The default value is 1.

### 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](#) added to the model, choose a **Distribution function** for the initial value of the auxiliary dependent variables and whether the initial value of the auxiliary dependent variables should be a scalar value or a distribution function.

The number of particles simulated can increase substantially and the following options are available for each of the [Auxiliary Dependent Variables](#) added to the model.

When **None** is selected, enter an initial value ( $rp_0$ ).

For the initial value of the auxiliary dependent variables, select **Normal** to sample from a normal distribution function, **Lognormal** to sample from a log-normal distribution function, or **Uniform** to sample from a uniform distribution function. For any selection, the **Number of values** sets the number of points in the distribution function. Enter a user-defined **Mean** (default 0) and **Standard deviation** (default 1). Select **List of values** to enter a list of numerical values for the auxiliary dependent variable directly.

By default auxiliary dependent variables are initialized after all other degrees of freedom. Select the **Initialize before particle momentum** check box to compute the initial value of the auxiliary dependent variable immediately after computing the initial particle position. By selecting this check box it is possible to define the initial particle velocity as a function of the auxiliary dependent variables. The order of initialization for the auxiliary dependent variables is described in more detail in [Initialization of Auxiliary Dependent Variables](#) in the [Particle Tracing Modeling](#) chapter.

#### ADVANCED SETTINGS

This section is shown if **Newtonian** or **Newtonian, first order** is selected from the **Formulation** list in the physics interface **Particle Release and Propagation** section.

By default the **Subtract moving frame velocity from initial particle velocity** check box is cleared. Select this check box to offset the initial velocity of released particles if the frame is moving. The frame velocity is taken from an active instance of the [Rotating Frame](#) feature if it has been added to the model. Thus, if the particle trajectories are computed in a rotating frame, clearing this check box means that the particles are released with respect to the rotating frame; selecting the check box means that the particle velocities are taken with respect to the non-rotating (inertial) frame.

#### *Axial Symmetry*

---

An **Axial Symmetry** node is automatically added to 2D axisymmetric components. The options available for the feature are exactly the same as for [Wall](#), except the **Boundary Selection** is locked and is only applicable on the symmetry axis. The [Accumulator \(Boundary\)](#) and/or [Secondary Emission](#) subnodes are available from the context menu (right-click the parent node) or from the **Physics** toolbar, **Attributes** menu.

## *Inlet*

---

Use the **Inlet** node to release particles into the modeling domain from selected boundaries. The **Nonlocal Accumulator** subnode is available from the context menu (right-click the parent node) or from the **Physics** toolbar, **Attributes** menu.

Go to **Release** for information about the following sections: **Release Times**, **Release Current Magnitude**, **Mass Flow Rate**, **Inherit Properties**, **Initial Particle Temperature**, **Initial Particle Mass**, **Initial Multiplication Factor**, **Initial Value of Auxiliary Dependent Variables**, and **Advanced Settings**.

### **COORDINATE SYSTEM SELECTION**

It is possible to specify the initial particle velocity in terms of the global coordinates or in another coordinate system defined for the model Component. Select an option from the **Coordinate system** list. By default **Global coordinate system** is selected. If other coordinate systems are defined, they can also be selected from the list. When specifying the initial particle velocity (see the **Initial Velocity** section), velocity or direction components can be specified using the basis vectors of whichever coordinate system has been selected from the list.

When a coordinate system other than **Global coordinate system** is selected from the **Coordinate system** list, arrows will appear in the Graphics window to indicate the orientation of the basis vectors of the coordinate system on the selected boundaries.

### **INITIAL POSITION**

Select an **Initial position**—**Mesh based** (the default), **Uniform distribution** (2D components) **Projected plane grid** (3D components), **Density**, or **Random**. **Mesh Based**, **Density**, and **Random** have the same settings as described for the **Release** node.



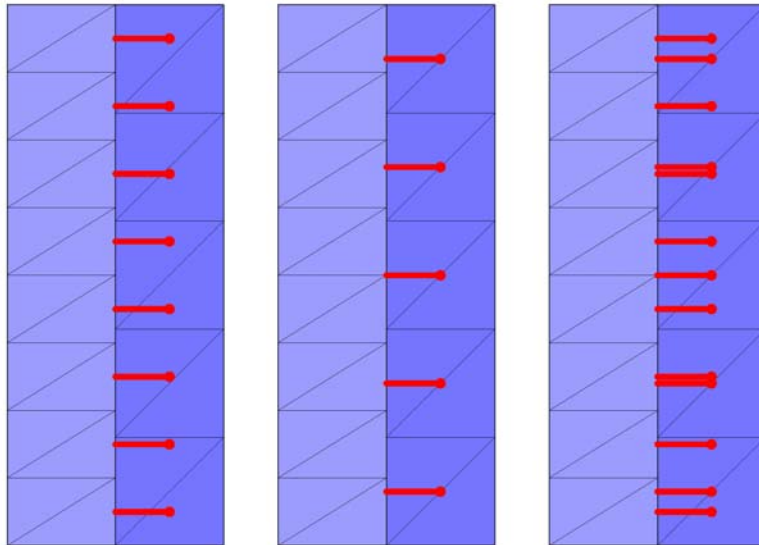
For 2D components, if **Uniform distribution** is selected, enter the **Number of particles per release**  $N$  (dimensionless). The default is 1. The union of the selected edges is divided into  $N$  segments of approximately equal length, and a particle is placed in the middle of each segment.

---



For 3D components, if **Projected plane grid** is selected, enter the **Number of particles per release**  $N$  (dimensionless). The default is 1. The particles are distributed on a plane grid in planes that are approximately tangential to the selected boundaries (for a plane boundary, you thus get a uniform distribution).

For inlets defined on identity pairs in an assembly, select an option from the **Release particles from boundaries** list: **Source** (the default), **Destination**, or **Source and Destination**. This option determines which boundary mesh to use when determining the initial particle positions. The mesh on either side of an inlet pair can be quite different, as shown in [Figure 3-2](#).



*Figure 3-2: Mesh-based release of particles from a source boundary (left), destination boundary (middle), or both source and destination (right). In each rectangle, the source boundary is on the side of the lighter-colored mesh.*

## INITIAL VELOCITY

Select an **Initial velocity**: **Expression** (the default), **Kinetic energy and direction**, **Constant speed, hemispherical**, **Constant speed, cone**, or **Constant speed, Lambertian** (3D only).

- For **Expression** enter expressions for the **Initial particle velocity**  $\mathbf{v}_0$  (SI unit: m/s) based on space dimension. Select the **Specify tangential and normal vector components** check box to enter expressions for the initial velocity in terms of the tangential and normal directions at the surface.



When using [The Particle Tracing for Fluid Flow Interface](#), enter a value or expression for the **Velocity field**  $\mathbf{u}$ . If another physics interface is present which computes the velocity field then this can be selected from the list.

- For **Kinetic energy and direction** enter the **Initial kinetic energy**  $E_0$  (SI unit: J). The default is 1 keV. Then enter coordinates for the **Initial particle direction**  $\mathbf{L}_0$  (dimensionless) based on space dimension. It is not necessary to normalize the components of the initial direction vector because this is done automatically when computing the initial velocity. Select the **Specify tangential and normal vector components** check box to enter expressions for the initial direction in terms of the tangential and normal directions at the surface.



The option **Kinetic energy and direction** is not supported for [The Particle Tracing for Fluid Flow Interface](#).

- For **Constant speed, hemispherical** enter a **Speed**  $v_0$  (SI unit: m/s), the **Number of particles in velocity space**  $N_{\text{vel}}$  (dimensionless), and the **Hemisphere axis**  $\mathbf{r}$ . Select the **Specify tangential and normal vector components** check box to enter expressions for the hemisphere axis in terms of the tangential and normal directions at the surface.
- For **Constant speed, cone** enter a **Speed**  $v_0$  (SI unit: m/s), the **Number of particles in velocity space**  $N_{\text{vel}}$  (dimensionless), the **Cone axis**  $\mathbf{r}$ , and the **Cone angle**  $\alpha$  (SI unit: rad). Select the **Specify tangential and normal vector components** check box to enter expressions for the cone axis in terms of the tangential and normal directions at the surface.
- The **Constant speed, Lambertian** option is only available in 3D. Enter a **Speed**  $v_0$  (SI unit: m/s), the **Number of particles in velocity space**  $N_{\text{vel}}$  (dimensionless), and the **Hemisphere axis**  $\mathbf{r}$ . The velocity distribution is a hemisphere in which the distribution of polar angles follows the cosine law, similar to the **Diffuse scattering** option for the [Wall](#) feature. Select the **Specify tangential and normal vector**



**components** check box to enter expressions for the hemisphere axis in terms of the tangential and normal directions at the surface.



When the **Specify tangential and normal velocity components** or **Specify tangential and normal vector components** check box is selected, arrows indicating the normal direction on the selected boundaries will appear in the Graphics window.

Note that the normal direction may be opposite the built-in variable for the boundary normal (for example,  $n_x$ ,  $n_y$ , and  $n_z$ ) to ensure that a positive value causes particles to be released into the simulation domain. This often occurs when the **Inlet** is applied to exterior boundaries.

When the normal direction used by the **Inlet** feature is opposite the normal vector defined by the geometry, the tangential directions are similarly inverted to ensure that the boundary coordinate system is right-handed.

For **Constant speed, hemispherical, Constant speed, cone,** and **Constant speed, Lambertian**, select an option from the **Sampling from distribution** list: **Deterministic** (the default) or **Random**. If **Deterministic** is selected, the initial velocity is computed by sampling from the velocity distribution in a deterministic and reproducible manner. If **Random** is selected, the particle velocity is sampled from the distribution using pseudo-random numbers and may not always generate the same results.

For **Constant speed, spherical, Constant speed, hemispherical, Constant speed, cone,** and **Constant speed, Lambertian**, select the **Allow nonuniform speeds in distribution** check box to allow particles from the same release point to have different initial speeds. This check box has no effect on the initial velocity directions of the released particles, only their velocity magnitudes. If this check box is selected, expressions for the initial speed in terms of the particle index can return a unique value for each particle in the distribution.



For example, if **Constant speed, hemispherical** is selected from the **Initial velocity** list, the **Allow nonuniform speeds in distribution** check box is selected, and the expression for the initial speed is  $2[\text{m/s}] * (\text{random}(\text{pt.pid}x) + 0.5)$ , then particles will be released with directions sampled from an isotropic hemisphere and with speeds sampled pseudorandomly between 0 and 2 m/s.



### *Nonlocal Accumulator*

Use the **Nonlocal Accumulator** subnode to communicate information from a particle's current position to the surface from which it was released.

The subnode is available from the context menu (right-click the **Inlet** parent node) or from the **Physics** toolbar, **Attributes** menu.

Each **Nonlocal accumulator** subnode defines a variable, called the accumulated variable, that is computed using variables defined on particles released by the parent **Inlet** node or on domains and boundaries encountered by such particles.

#### **ACCUMULATOR SETTINGS**

Select an **Accumulator type**: **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 accumulated variable is proportional to the instantaneous value of the **Source** term  $R$  for all applicable particles.
- For **Elements and time** the time derivative of the accumulated variable is proportional to the instantaneous value of the **Source** term  $R$  for all applicable particles, and thus the accumulated variable considers the time history of particles in the modeling domain instead of just their current values.

Enter the **Accumulated variable name**. The default is `rp1`.

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.

Select a **Source geometric entity level** — **Domains**, **Boundaries**, or **Domains and boundaries**.

- If **Domains** is selected, particles only contribute to the accumulated variable on their releasing surface if they are still active; that is, they are still propagating through a domain.
- If **Boundaries** is selected, the particles only contribute to the accumulated variable if they have become stuck or frozen to a boundary somewhere in the model.
- If **Domains and boundaries** is selected, all of the active, stuck, and frozen particles released by a feature can contribute to the accumulated variable.

### 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 variable is piecewise discontinuous across different boundary elements. Select the **Compute smoothed accumulated variable** check box to define a spatially smoothed accumulated variable; at each point on the boundary, the smoothed accumulated variable is the average of the accumulated variable over a circle. Enter a **Smoothing radius**  $r$  (SI unit: m). The default is 0.1 m.

### *Outlet*

---

Use the **Outlet** node to determine what should happen to particles when exiting the modeling domain. The **Accumulator (Boundary)** subnode is available from the context menu (right-click the parent node) or from the **Physics** toolbar, **Attributes** menu.

### OUTLET

Select a **Wall condition** — **Freeze** (the default) or **Disappear**. When **Freeze** is selected, the particle position and velocity remains frozen once the particle has made contact with the outlet boundary. This allows information about the particle velocity, energy and so forth to be recovered at the instant the particles left the modeling domain. For

**Disappear** it means that the particles are not visualized once they strike the outlet boundary.



- [About the Outlet Boundary Conditions](#)
- [Theory for the Mathematical Particle Tracing Interface](#)

---

### *Particle Continuity*

---

Use the **Particle Continuity** node to specify that particles should cross a pair boundary as if it were invisible. Pair boundaries appear when the geometry sequence ends in **Form Assembly** instead of **Form Union**. Such boundaries require special handling because the mesh elements on either side of the pair boundary aren't required to match up exactly.

#### **PAIR SELECTION**

Select one or more identity pairs to allow particles to cross between the source and destination boundaries of these pairs. Such identity pairs are typically created automatically on interior boundaries when the geometry sequence ends in a **Form Assembly** node instead of a **Form Union** node.



The **Particle Continuity** node does not cause the particle position components to change discontinuously; the source and destination boundaries for the identity pairs must be overlapping.

If the geometry sequence ends in a **Form Union** node, the **Particle Continuity** node usually isn't necessary because particles can freely cross interior boundaries where no boundary condition has been applied.

---

### *Auxiliary Dependent Variable*

---

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



- [Auxiliary Dependent Variables and Residence Time](#)
  - [Initialization of Auxiliary Dependent Variables](#)
-

### 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** chose whether to integrate the equation you have defined **With respect to time** or **Along particle trajectory**. For example, to compute the residence time of a group of particles in a given system, set the **Source** to 1 and set **Integrate** to **With respect to time**. To compute the length of the particle trajectory, set the **Source** to 1 and set **Integrate** to **Along particle 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  $mol/m^3$ .

### *Release from Edge*

---

Use the **Release from Edge** node to release particles into the modeling domain from a set of selected edges in a 3D geometry.

Go to [Release](#) for information about the following sections: **Release Times**, **Release Current Magnitude**, **Mass Flow Rate**, **Initial Velocity**, **Inherit Properties**, **Initial Particle Temperature**, **Initial Particle Mass**, **Initial Multiplication Factor**, **Initial Value of Auxiliary Dependent Variables**, and **Advanced Settings**.

### INITIAL POSITION

Select an **Initial position**—**Mesh based** (the default), **Density**, or **Random**. [Mesh Based](#), [Density](#), and [Random](#) have the same settings as described for the [Release](#) node. If **Uniform distribution** is selected, enter the **Number of particles per release  $N$**  (dimensionless). The union of the selected edges is divided into  $N$  segments of approximately equal length, and a particle is placed in the middle of each segment.

### *Release from Point*

---

Use the **Release from Point** node to release particles into the modeling domain from selected points in the geometry. By default, one particle is released at each selected

point, although a larger number of particles will be released if the initial velocity or an auxiliary dependent variable is sampled from a distribution of values.

Go to [Release](#) for information about the following sections: **Release Times**, **Release Current Magnitude**, **Mass Flow Rate**, **Initial Velocity**, **Inherit Properties**, **Initial Particle Temperature**, **Initial Particle Mass**, **Initial Multiplication Factor**, **Initial Value of Auxiliary Dependent Variables**, and **Advanced Settings**.


### *Release from Grid*

---

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

Go to [Release](#) for information about the following sections: **Release Times**, **Release Current Magnitude**, **Mass Flow Rate**, **Initial Velocity**, **Inherit Properties**, **Initial Particle Temperature**, **Initial Particle Mass**, **Initial Multiplication Factor**, **Initial Value of Auxiliary Dependent Variables**, and **Advanced Settings**.

#### **INITIAL COORDINATES**

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

Select an option from the **Grid type** list: **All combinations** (the default) or **Specified combinations**. If **Specified combinations** is selected, the number of initial coordinates entered for each space dimension must be equal, and the total number of particles released is equal to the length of one of the lists of initial coordinates. If **All combinations** is selected, the total number of particles 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 particles will be released, including every possible combination of the initial  $x$ - and  $y$ -coordinates. If **Specified combinations** is selected, 4 particles will be released with initial positions (0,2), (1,4), (2,6), and (3,8).

The number of released particles may increase if the initial velocity or initial values of auxiliary dependent variables involve sampling from a distribution, in which case a

large number of particles may be released at each initial position. The position for any particles with initial coordinates outside the geometry are set to not-a-number (NaN), so the particles do not appear when plotted during postprocessing.

### *Override Properties*

---

By default, all particles are assigned properties such as mass, charge number, and Lagrangian based on the settings entered for [Particle Properties](#). Add an **Override Properties** node to create alternative sets of properties that can be assigned to the particles. The user inputs in the **Settings** window for **Override Properties** are identical to those in the **Settings** window for **Particle Properties**. The properties in an **Override Properties** node can then be applied to particles using the **Inherit Properties** section in the **Release**, **Inlet**, **Release from Grid**, and **Secondary Emission** features.



### Overriding Particle Properties

---

### *Release from Data File*

---

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

Go to [Release](#) for information about the following sections: **Release Times**, **Release Current Magnitude**, **Mass Flow Rate**, **Inherit Properties**, **Initial Particle Temperature**, **Initial Multiplication Factor**, and **Advanced Settings**.

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

The imported data file should be a text file (\*.txt) arranged in a spreadsheet format; that is, each row corresponds to a distinct particle 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 particles at the positions (0.1, 0.2, 0.6) and (0.2, 0.4, 0.8) in a three-dimensional geometry:

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

### INITIAL POSITION

**Browse** your computer 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 particle position vectors. The default value, 0, indicates the first column.

### INITIAL VELOCITY

Select an option from the **Initial velocity** list: **Expression** (the default), **From file**, **Kinetic energy and direction**, **Constant speed, spherical**, **Constant speed, hemispherical**, **Constant speed, cone**, **Constant speed, Lambertian** (3D only), or **Maxwellian**.

- For **From file**, enter the **Index of first column containing velocity data  $i$** . The default is 3. The columns are zero-indexed; that is, an index of 0 corresponds to the first column.
- For all other options, the settings are the same as for the [Release](#) node.

### INITIAL PARTICLE MASS

This section is only available for [The Particle Tracing for Fluid Flow Interface](#) and is only shown when the **Compute particle mass** check box is selected in the physics interface **Additional Variables** section.

The initial particle mass can be a single value or can be sampled from a distribution. Select an option from the **Distribution function** list—**From file**, **None** (the default), **Normal**, **Lognormal**, **Uniform**, or **List of values**.

- When **From file** is selected, enter the **Index of column containing mass data**. The default value is 3. The columns are zero-indexed; that is, an index of 0 corresponds to the first column.
- For all other options, the settings are the same as for the [Release](#) node.

### 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 active **Auxiliary Dependent Variable** nodes in the model, choose an option from the **Distribution function** list: **From file**, **None** (the default), **Normal**, **Lognormal**, **Uniform**, or **List of Values**.

- When **From file** is selected, enter the **Index of column containing data**. The default value is 3. The columns are zero-indexed; that is, an index of 0 corresponds to the first column.
- For all other options, the settings are the same as for the [Release](#) node..

# Theory for the Mathematical Particle Tracing Interface

The equation system solved by the Mathematical Particle Tracing interface depends on the formulation selected. In the following, the particle has a position vector  $\mathbf{q}$ , a velocity vector  $\mathbf{v}$ , and a momentum vector  $\mathbf{p}$ .

The [Mathematical Particle Tracing Interface](#) theory is described in this section:

- [Newtonian Formulation](#)
- [Hamiltonian Formulation](#)
- [Lagrangian Formulation](#)
- [Massless Formulation](#)
- [Particle Tracing in Rotating Frames](#)
- [Initial Conditions — Position](#)
- [Initial Conditions — Velocity](#)
- [Particle-Particle Interactions](#)
- [Auxiliary Dependent Variables](#)
- [About the Boundary Conditions for the Particle Tracing Interfaces](#)
- [Accumulator Theory: Domains](#)
- [Accumulator Theory: Boundaries](#)
- [References for the Mathematical Particle Tracing Interface](#)

## *Newtonian Formulation*

---

The Mathematical Particle Tracing interface makes it possible to solve ordinary differential equations for the particle positions. When the Newtonian formulation is selected, the particle position is computed using Newton's second law:

$$\frac{d}{dt}(m_p \mathbf{v}) = \mathbf{F}$$

where  $m_p$  is the particle mass (SI unit: kg),  $\mathbf{v}$  is the particle velocity (SI unit: m/s), and  $\mathbf{F}$  is the force exerted on the particle (SI unit: N). The force is either specified directly or using a susceptibility multiplied by a suitable field:

$$\mathbf{F} = \boldsymbol{\chi} \cdot \boldsymbol{\Gamma}$$

where  $\boldsymbol{\chi}$  is the susceptibility tensor and  $\boldsymbol{\Gamma}$  is a vector field. The particle velocity is defined as:

$$\mathbf{v} = \frac{d\mathbf{q}}{dt}$$

where  $\mathbf{q}$  is the particle position vector (SI unit: m).

When using a relativistic correction of the particle mass for particles with very high velocity, the particle mass  $m_p$  is defined as

$$m_p = \frac{m_r}{\sqrt{1 - \mathbf{v} \cdot \mathbf{v}/c^2}}$$

where  $m_r$  is the rest mass (SI unit: kg) and  $c = 2.99792458 \times 10^8$  m/s is the speed of light in a vacuum, which is a built-in physical constant.

If the default **Newtonian** formulation is selected, Newton's second law is expressed as a set of second-order ordinary differential equations for the particle position vector components:

$$\frac{d}{dt} \left( m_p \frac{d\mathbf{q}}{dt} \right) = \mathbf{F}$$

If the first-order Newtonian formulation is selected, Newton's second law is instead expressed as a set of coupled first-order ordinary differential equations:

$$\frac{d}{dt} (m_p \mathbf{v}) = \mathbf{F}$$

$$\mathbf{v} = \frac{d\mathbf{q}}{dt}$$

#### USING FIRST-ORDER AND SECOND-ORDER FORMULATIONS

The advantage of using the first-order Newtonian formulation is that it avoids mixing first- and second-order equations when any [Auxiliary Dependent Variables](#) are also solved for, since the equations for auxiliary dependent variables are always first-order. This allows efficient, high-order explicit time stepping methods to be used for a wider class of problems compared to the second-order Newtonian formulation.

The explicit time stepping methods, such as Runge-Kutta methods, are most suitable for nonstiff problems. In a particle tracing context this often means that the particle position, velocity, or other dependent variables aren't expected to change abruptly over an extremely short time interval.

### *Hamiltonian Formulation*

---

Sometimes it is more convenient to describe the particle motion using a *Hamiltonian* formulation. In this case, the following system of first-order ODEs is solved:

$$\frac{d\mathbf{q}}{dt} = \mathbf{v}_H(\mathbf{q}, t)$$

$$\frac{d\mathbf{p}}{dt} = \mathbf{F}_H(\mathbf{q}, \mathbf{v}, t)$$

The particle velocity  $\mathbf{v}_H$  (SI unit: m/s) and particle force  $\mathbf{F}_H$  (SI unit: N) can be entered directly, or, more conveniently by specifying the Hamiltonian,  $H$  (SI unit: J). The Hamiltonian is related to the particle velocity and force via:

$$\mathbf{v}_H = \frac{\partial H}{\partial \mathbf{p}}$$

and

$$\mathbf{F}_H = -\frac{\partial H}{\partial \mathbf{q}}$$

For a more in-depth example, see [Choosing a Formulation](#) in the [Particle Tracing Modeling](#) chapter.

### *Lagrangian Formulation*

---

Instead of a Hamiltonian, a *Lagrangian* formulation can be used, in which case Lagrange's equation is solved:

$$\frac{d}{dt} \left( \frac{\partial L}{\partial \mathbf{v}} \right) = \frac{\partial L}{\partial \mathbf{q}}$$

Lagrange's equation is rearranged before solving into the following form:

$$\frac{\partial}{\partial t} \left( m \frac{\partial \mathbf{q}}{\partial t} \right) = N^{-1} \left( \frac{\partial L}{\partial \mathbf{q}} - \left( \mathbf{v} \cdot \nabla \frac{\partial L}{\partial \mathbf{v}} \right) \right)$$

where  $N$  is the Hessian of the Lagrangian with respect to the velocities:

$$N_{ij} = \frac{\partial^2 L}{\partial v_i \partial v_j}$$

For a more in-depth example, see [Choosing a Formulation](#) in the [Particle Tracing Modeling](#) chapter.

---

### *Massless Formulation*

The pathline of a massless particle is given by:

$$\frac{d\mathbf{q}}{dt} = \mathbf{v}$$

where  $\mathbf{v}$  is a user-defined velocity vector.

When the **Newtonian**, **Lagrangian**, or **Massless** formulations are used there are degrees of freedom for the particle position only. When the **Newtonian, first order** formulation is used there are degrees of freedom for the particle position and velocity. When the **Hamiltonian** formulation is used, there are degrees of freedom for the particle position and momentum.

---

### *Particle Tracing in Rotating Frames*

The [Rotating Frame](#) feature exerts fictitious forces on particles that are moving in a rotating frame of reference. Optionally, it also applies an offset to the initial velocity of released particles, depending on whether their velocity is initialized with respect to the inertial (nonaccelerating) frame or the rotating frame.

Consider a particle of mass  $m_p$  (SI unit: kg) moving in a noninertial frame of reference with translational acceleration  $\mathbf{W}$  (SI unit: m/s<sup>2</sup>) that is rotating at angular velocity  $\mathbf{\Omega}$  (SI unit: rad/s) about a center of rotation with position  $\mathbf{r}_{bp}$  (SI unit: m). Note that  $\mathbf{\Omega}$  is a vector quantity that also indicates the orientation of the axis of rotation and the sense of rotation (clockwise or counterclockwise), according to the standard right-hand rule; that is, if your right thumb points in the direction of the angular velocity vector, your fingers sweep in the direction of rotation. The total fictitious force  $\mathbf{F}_{fr}$  (SI unit: N) exerted on a particle in this noninertial frame of reference is ([Ref. 1](#))

$$\mathbf{F}_{\text{fr}} = \mathbf{F}_{\text{cen}} + \mathbf{F}_{\text{cor}} + \mathbf{F}_{\text{eul}} + \mathbf{F}_{\text{acc}}$$

where  $\mathbf{F}_{\text{cen}}$  is the centrifugal force,

$$\mathbf{F}_{\text{cen}} = m_p \boldsymbol{\Omega} \times (\mathbf{r} \times \boldsymbol{\Omega})$$

$\mathbf{F}_{\text{cor}}$  is the Coriolis force,

$$\mathbf{F}_{\text{cor}} = 2m_p \mathbf{v} \times \boldsymbol{\Omega}$$

$\mathbf{F}_{\text{eul}}$  is the Euler force,

$$\mathbf{F}_{\text{eul}} = m_p \mathbf{r} \times \dot{\boldsymbol{\Omega}}$$

and  $\mathbf{F}_{\text{acc}}$  is the force resulting from linear acceleration of the frame,

$$\mathbf{F}_{\text{acc}} = -m_p \mathbf{W}$$

The displacement  $\mathbf{r}$  (SI unit: m) of the particle is defined with respect to the center of rotation,

$$\mathbf{r} = \mathbf{q} - \mathbf{r}_{\text{bp}}$$

where  $\mathbf{q}$  (SI unit: m) is the particle position. The **Rotating Frame** feature only accounts for rotational motion of the frame, not translational motion, so the term  $\mathbf{F}_{\text{acc}}$  is canceled.

When the **Rotating Frame** feature is active, it is possible to subtract the frame velocity from the initial particle velocity when releasing particles. This is equivalent to specifying the initial particle velocity with respect to the inertial frame. The relationship between the initial velocity in the inertial frame  $\mathbf{v}_{i,\text{in}}$  and in the rotating frame  $\mathbf{v}_{i,\text{rot}}$  is

$$\mathbf{v}_{i,\text{in}} = \mathbf{v}_{i,\text{rot}} - \mathbf{v}_f$$

where  $\mathbf{v}_f$  is the frame velocity at the particle's position,

$$\mathbf{v}_f = \boldsymbol{\Omega} \times \mathbf{r}$$

### *Initial Conditions — Position*

---

Initial conditions for the particle position are either mesh based, from a user-specified grid, uniformly distributed or based on an analytic expression.

## Initial Conditions — Velocity

---

There are several options available for specifying the initial velocity of particles:

### EXPRESSION

The default is to specify an initial expression for each component of the velocity,  $\mathbf{v}_0$ . In the case of the Newtonian, first-order Newtonian, and Lagrangian formulations the following condition is implemented:

$$\frac{d\mathbf{q}}{dt} = \mathbf{v}_0$$

In this case the initial velocity of every particle is defined using the same expression. The particles may have different initial velocity if  $\mathbf{v}_0$  is a function of initial position.

### KINETIC ENERGY AND DIRECTION

If the initial kinetic energy  $E_0$  (SI unit: J) and initial particle direction  $\mathbf{L}_0$  (dimensionless) are specified, for the Newtonian, first-order Newtonian, and Lagrangian formulations the following condition is implemented:

$$\frac{d\mathbf{q}}{dt} = \frac{\mathbf{L}_0}{|\mathbf{L}_0|} \sqrt{\frac{2E_0}{m_p}}$$

where  $m_p$  is the mass of the particle. If either Newtonian formulation is used and a relativistic correction is enabled, instead the following condition is implemented:

$$\frac{d\mathbf{q}}{dt} = \frac{c\mathbf{L}_0}{|\mathbf{L}_0|} \sqrt{1 - \frac{1}{\left(\frac{E_0}{m_r c^2} + 1\right)^2}}$$

where  $m_r$  (SI unit: kg) is the rest mass of the particle and  $c = 2.99792458 \times 10^8$  m/s is the speed of light in a vacuum.

### MAXWELLIAN

When a Maxwellian initial velocity distribution is selected, an array of  $N_v$  initial velocity values is created for each particle in each velocity direction. For a given initial temperature, in each velocity direction, the distribution function is given by:

$$f(v_i) = \sqrt{\frac{m_p}{2\pi k_B T_0}} \exp\left(-\frac{m_p v_i^2}{2k_B T_0}\right)$$

where

- $v_i$  (SI unit: m/s) is the initial velocity component in the  $i^{\text{th}}$  direction,
- $m_p$  (SI unit: kg) is the particle mass,
- $T_0$  (SI unit: K) is the temperature, and
- $k_B = 1.3806488 \times 10^{-23}$  J/K is the Boltzmann constant.

So the total distribution function is:

$$\mathbf{f}(\mathbf{v}) = \prod_{i=1}^{\text{nsdim}} f(v_i)$$

The probability a particle has an initial velocity  $v_i$  is given by:

$$\mathbf{g}(\mathbf{v}) = \prod_{i=1}^{\text{nsdim}} v_i f(v_i)$$

This can clearly generate a large number of particles because the total number of initial velocities are  $N_v^{\text{nsdim}}$ . This option is particularly useful in plasma modeling, AC/DC modeling, and in Monte Carlo simulations for molecular dynamics applications.

### CONSTANT SPEED, SPHERICAL

Sometimes it is desirable to release  $N_{\text{vel}}$  particles with the same speed isotropically in velocity space. Defining the speed as  $c$ , the following expressions generate such a velocity distribution in 2D according to:

$$\begin{aligned} v_x &= c \cos \theta \\ v_y &= c \sin \theta \end{aligned}$$

where  $\theta$  goes from 0 to  $2\pi$  in  $N_{\text{vel}}$  steps.

In 3D the velocity distribution is given by:

$$\begin{aligned} v_x &= c \cos \theta \sin \varphi \\ v_y &= c \sin \theta \sin \varphi \\ v_z &= c \cos \varphi \end{aligned}$$

The azimuthal angle  $\theta$  is uniformly distributed from 0 to  $2\pi$ . The polar angle  $\varphi$  is sampled from the interval  $[0, \pi]$  with probability density proportional to  $\sin \varphi$ . The



polar angle is arbitrarily chosen as the angle that the initial velocity makes with the positive  $z$ -axis, but any direction could be chosen because the sphere is isotropic.

#### **CONSTANT SPEED, HEMISPHERICAL**

The **Constant speed, hemispherical** option is the same as the **Constant speed, spherical** option, except that in 2D  $\theta$  goes from 0 to  $\pi$  and in 3D  $\varphi$  goes from 0 to  $\pi/2$ . The angle ( $\theta$  in 2D or  $\varphi$  in 3D) is measured from the direction given by the **Hemisphere axis** setting.

#### **CONSTANT SPEED, CONE**

The **Constant speed, cone** option is the same as the **Constant speed, spherical** option, except that in 2D  $\theta$  goes from 0 to  $\alpha$  and in 3D  $\varphi$  goes from 0 to  $\alpha$ . The angle ( $\theta$  in 2D or  $\varphi$  in 3D) is measured from the direction given by the **Cone axis** setting.

#### **CONSTANT SPEED, LAMBERTIAN**

The **Constant speed, Lambertian** option releases particles within a hemisphere in 3D velocity space, but the probability distribution function is different from that of the **Constant speed, hemisphere** option. Recall that for an isotropic hemispherical distribution the polar angle  $\varphi$  has a probability density proportional to  $\sin \varphi$ ; for the Lambertian distribution the probability density is instead proportional to  $\sin \varphi \cos \varphi$ . Because of this extra cosine term, distributions following this probability density are said to follow Lambert's cosine law or (in molecular dynamics) Knudsen's cosine law.

### *Particle-Particle Interactions*

---

#### **COULOMB FORCE**

The Coulomb force on particle  $i$  in a system of  $N$  particles is defined as:

$$\mathbf{F}_i = \frac{e^2}{4\pi\epsilon_0} \sum_{j=1}^N Z_i Z_j \frac{(\mathbf{r}_i - \mathbf{r}_j)}{|\mathbf{r}_i - \mathbf{r}_j|^3}$$

where

- $e = 1.602176565 \times 10^{-19}$  C is the elementary charge,
- $\epsilon_0 = 8.854187817 \times 10^{-12}$  F/m is the permittivity of vacuum,
- $\mathbf{r}_i$  is the position vector of the  $i^{\text{th}}$  particle (SI unit: m),

- $\mathbf{r}_j$  is the position vector of the  $j^{\text{th}}$  particle (SI unit: m), and
- $Z$  (dimensionless) is the particle charge number.

The Coulomb force is repulsive for particles of the same charge and attractive for particles with opposite charge.

### LENNARD-JONES FORCE

The Lennard-Jones interaction is often used as the intermolecular potential function to estimate the transport properties of a gas. The Lennard-Jones potential is given by:

$$U(r) = 4\epsilon \left[ \left( \frac{\sigma}{r} \right)^{12} - \left( \frac{\sigma}{r} \right)^6 \right]$$

where  $r$  is the distance between the particles (SI unit: m),  $\epsilon$  is the interaction strength (SI unit: N·m), and  $\sigma$  is the collision diameter (SI unit: m).

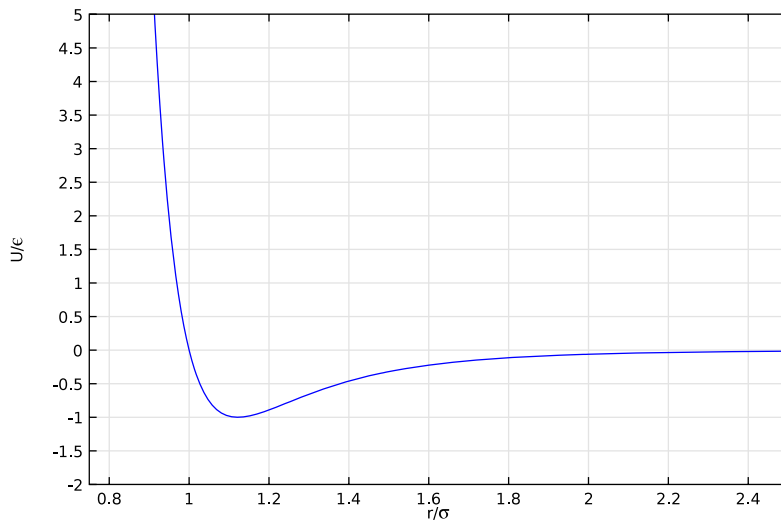


Figure 3-3: Plot of the Lennard-Jones potential normalized to the interaction strength versus the nondimensional radial distance.

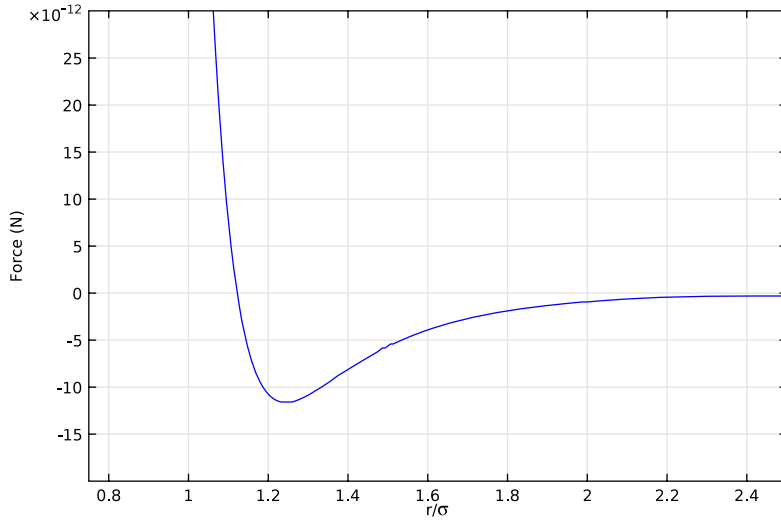
Using the fact that the force is computed as:

$$\mathbf{F} = -\nabla U$$

the expression for the force on the  $i^{\text{th}}$  particle becomes:

$$\mathbf{F}_i = \frac{24\epsilon}{\sigma} \sum_{j=1}^N \left[ 2 \left( \frac{\sigma}{|\mathbf{r}_i - \mathbf{r}_j|} \right)^{13} - \left( \frac{\sigma}{|\mathbf{r}_i - \mathbf{r}_j|} \right)^7 \right] \frac{(\mathbf{r}_i - \mathbf{r}_j)}{|\mathbf{r}_i - \mathbf{r}_j|}$$

A typical interaction strength is around  $1.6 \times 10^{-21}$  J and a typical collision diameter is around  $3.3 \times 10^{-10}$  m (3.3 Å). The first term in the square brackets is a repulsive force due to internuclear repulsion, and the second term is an attractive force known as the *London dispersion force* (LDF). The variation of the force with distance is plotted in [Figure 3-4](#).



*Figure 3-4: Plot of the Lennard-Jones force versus nondimensionalized radial distance between particles.*

#### LINEAR ELASTIC FORCE

The linear elastic force  $\mathbf{F}_i$  on particle  $i$  in a system of  $N$  particles is defined as

$$\mathbf{F}_i = -k_s \sum_{j=1; i \neq j}^N (|\mathbf{r}_i - \mathbf{r}_j| - r_0) \frac{(\mathbf{r}_i - \mathbf{r}_j)}{|\mathbf{r}_i - \mathbf{r}_j|}$$

where  $k_s$  is the spring constant (SI unit: N/m) and  $r_0$  is the equilibrium distance between particles (SI unit: m).

### *Auxiliary Dependent Variables*

---

In certain cases it is desirable to add additional degrees of freedom to compute particle mass, temperature, size, spin, and so forth. The additional degree of freedom is solved for each particle. For each degree of freedom,  $w$ , the following ODE is solved for each particle:

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

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

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

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

### *About the Boundary Conditions for the Particle Tracing Interfaces*

---

The equations of motion for the particle trajectories are supplemented with a variety of options to describe how the particles behave when they contact surfaces in the geometry.

#### **ABOUT THE WALL BOUNDARY CONDITIONS**

The following nomenclature is used with time as an example:

- Value of a variable precontact,  $w$ .
- Value of a variable at the moment of contact,  $w_c$ .
- Value of a variable postcontact,  $w'$ .

When a particle comes in contact with a wall, at time  $t_c$ , the following options are available for what happens as a result of the particle-wall interaction.

#### *Disappear*

When the particle strikes the wall it disappears from view. Mathematically the particle location is set to not-a-number (NaN) at all time steps after the particle makes contact with the wall:

$$\mathbf{q}' = \text{NaN}$$

This condition means that the particle is not rendered during results processing.

### *Freeze*

The particle position and velocity remain the same at all time steps  $t > t_c$ :

$$\mathbf{q}' = \mathbf{q}_c$$

$$\mathbf{v}' = \mathbf{v}_c$$

Because the velocity is frozen at the time of impact, this boundary condition allows for recovery of velocity and energy distribution functions.

### *Stick*

The particle follows the motion of the wall after contact:

$$\mathbf{q}' = \mathbf{q}_w$$

$$\mathbf{v}' = \mathbf{v}_w$$

### *Bounce*

The particle is reflected from the wall in the tangent plane. The incident angle and reflected angle are the same with respect to the surface normal:

$$\mathbf{q}' = \mathbf{q}$$

$$\mathbf{v}' = \mathbf{v} - 2(\mathbf{n} \cdot \mathbf{v})\mathbf{n}$$

This conserves energy if the Hamiltonian depends isotropically on momentum.

### *Diffuse Scattering*

The particle is reflected from the wall with a velocity vector according to Knudsen's cosine law. Specifically, for the particle position:

$$\mathbf{q}' = \mathbf{q}$$

and for the particle velocity in 3D:

$$v_{t1} = |\mathbf{v}_c| \sin \theta \sin \phi$$

$$v_{t2} = |\mathbf{v}_c| \sin \theta \cos \phi$$

$$v_n = |\mathbf{v}_c| \cos \theta$$

where  $\mathbf{v}_c$  is the velocity vector when the particle strikes the wall  $v_{t1}$  and  $v_{t2}$  are the outgoing tangential components of the velocity and  $v_n$  is the normal component of

the velocity. The angle  $\phi$  is a uniformly distributed random number between 0 and  $2\pi$ . The angle  $\theta$  is defined as:

$$\theta = \text{asin}(\sqrt{\Gamma})$$

where  $\Gamma$  is a uniformly distributed random number between 0 and 1. The probability distribution of the normal velocity component is directly proportional to  $\cos(\theta)$ .

In 2D, the tangential and normal velocity components are:

$$v_t = |\mathbf{v}_c| \sin \theta$$

$$v_n = |\mathbf{v}_c| \cos \theta$$

where the angle  $\theta$  is defined as:

$$\theta = \text{acos}(\Gamma) - \frac{\pi}{2}$$

where  $\Gamma$  is now a uniformly distributed random number between -1 and 1.

#### *Mixed Diffuse and Specular Reflection*

The particle has probability  $\gamma$  to be reflected specularly, as if using the **Bounce** condition. Otherwise the particle is reflected diffusely, as if using the **Diffuse scattering** condition.

#### *General Reflection*

The general reflection option allows for arbitrary velocities to be specified for the particles after they strike the wall:

$$\mathbf{q}' = \mathbf{q}$$

and

$$\mathbf{v}' = \mathbf{v}_p$$

where  $\mathbf{v}_p$  (SI unit: m/s) is the user-defined velocity vector. The velocity can be specified either in Cartesian coordinates or in the normal-tangent coordinate system.

### **ABOUT THE INLET BOUNDARY CONDITIONS**

At the inlet the number of particles, particle position, initial velocity, and the number of releases is specified. An **Inlet** node can contribute with a **Wall** or **Outlet** node, so it is possible to specify a behavior for particles that return to the inlet at a later time.

## ABOUT THE OUTLET BOUNDARY CONDITIONS

At the outlets the particles either freeze or disappear.

### *Accumulator Theory: Domains*

---

The **Accumulator (Domain)** node is used to transfer information from particles 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 may be discontinuous between adjacent mesh elements.

The name of the accumulated variable is specified in the **Accumulated variable name** edit 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 particle'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 particles in that mesh element:

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

where  $N$  is the total number of particles in the element and  $V$  is the area or volume of the mesh element. In other words, the contribution of each particle to the accumulated

variable is distributed uniformly over the mesh element the particle is in, regardless of the particle'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 particles 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 particles in the mesh element, instead of the instantaneous positions of the particles. As each particle propagates, it will leave behind a trail based on its contributions to the accumulated variables in 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 particle 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 particles to the boundaries they hit or pass through. 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** edit 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 particle'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).

When **Particles in boundary elements** is selected from the **Accumulate over** list, the accumulated variable in a boundary element gets incremented by the source term  $R$  whenever a particle freezes or sticks to the boundary:

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

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 particle is uniformly distributed over the boundary element it freezes or sticks to.

If instead **Particle-wall interactions** is selected from the **Accumulate over** list, then the accumulated variable gets incremented regardless of what type of particle-wall interaction occurs. Thus, it is possible for the same particle 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 3-3: BUILT-IN GLOBAL VARIABLES FOR BOUNDARY ACCUMULATORS

NAME	DESCRIPTION
<scope>. <name>_ave	Average of accumulated variable
<scope>. <name>_int	Integral 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 the names the Accumulator and parent feature. For example, the average of the accumulated variable over a boundary may be called `pt.wall1.bacc1.rpb_ave`, where `pt` is the name of the Mathematical Particle Tracing interface, `wall1` 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 **Add/Replace Expression** menus during results evaluation.

These global variables are computed by defining a set of component 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 3-4: BUILT-IN GLOBAL VARIABLE DEFINITIONS FOR BOUNDARY ACCUMULATORS


NAME	EXPRESSION
<scope>.<name>_ave	<wscope>.aveop(<scope>.<name>)
<scope>.<name>_int	<wscope>.intop(<scope>.<name>)
<scope>.<name>_max	<wscope>.maxop(<scope>.<name>)
<scope>.<name>_min	<wscope>.minop(<scope>.<name>)
<scope>.<name>_sum	<wscope>.intop(<scope>.<name>/<scope>.meshVol)

Here, <wscope> is the scope of the parent boundary feature, for example, pt.wall1.

### *References for the Mathematical Particle Tracing Interface*

1. L.D. Landau and E.M. Lifshitz, *Mechanics*, 3rd ed., Elsevier, 1976.

# Charged Particle Tracing



This chapter describes the Charged Particle Tracing interface, which is found under the **AC/DC** branch (  ) when adding a physics interface.

In this chapter:

- [The Charged Particle Tracing Interface](#)
- [Theory for the Charged Particle Tracing Interface](#)

See [The Electromagnetics Interfaces](#) in the *COMSOL Multiphysics Reference Manual* for other AC/DC interface and feature node settings.

# The Charged Particle Tracing Interface

The **Charged Particle Tracing (cpt)** interface () found under the **AC/DC>Particle Tracing** branch () when adding a physics interface, is used to model charged particle orbits under the influence of electromagnetic forces. In addition, it can also model two-way coupling between the particles and fields. Some typical applications are particle accelerators, vacuum tubes and ion implanters. The physics interface supports time-domain modeling only in 2D and 3D.

The physics interface solves the equation of motion for charged particles. It is possible to add forces to ions and electrons, including predefined electric, magnetic, and collisional forces. Collisions can be modeled as a continuous force or using a Monte Carlo model in which particles can undergo elastic or inelastic collisions with a rarefied gas, including charge exchange and ionization reactions.

When this physics interface is added, these default nodes are also added to the **Model Builder: Wall** and **Particle Properties**. Then, from the **Physics** toolbar, add other nodes that implement, for example, boundary conditions and release features. You can also right-click **Charged Particle Tracing** to select physics features from the context menu.

## SETTINGS

The **Label** is the default physics interface name.

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 `cpt`.

## PARTICLE RELEASE AND PROPAGATION

The **Formulation**, **Relativistic Correction**, **Store extra time steps for wall interactions**, and **Maximum number of secondary particles** settings are the same as for [The Mathematical Particle Tracing Interface](#), except that only the **Newtonian**, **Newtonian, first order**, and **Massless** formulations are available.

### Particle Release Specification

Select a **Particle release specification**—**Specify release times** (the default) or **Specify current**. If **Specify release times** is selected, then each model particle is treated as the instantaneous position of one or more charged particles for the purpose of modeling particle-field interactions. This means, for example, that if the [Space Charge Density Calculation](#) node is used, the space charge density is only nonzero in mesh elements that are currently occupied by particles.

If **Specify current** is selected, then for the purpose of modeling particle-field interactions, each model particle traces a path that is followed by a number of charged particles per unit time. This means that the charge density computed by the [Space Charge Density Calculation](#) feature is nonzero in all mesh elements that the particle trajectories pass through, not just at the instantaneous positions of the particles. In other words, the particles leave behind a trail of space charge as they propagate.

The **Specify current** option is primarily used to model charged particle beams in which the charge and current density do not change over time. Changing the **Particle release specification** affects some inputs in the settings windows for release features such as the [Release](#) and [Inlet](#) nodes. In addition, the [Surface Charge Density](#) node is only available with the **Specify release times** option, while the [Current Density](#), [Heat Source](#), and [Etch](#) nodes are only available with the **Specify current** option.

### ADDITIONAL VARIABLES

Except for the option described below, these settings are the same as for [The Mathematical Particle Tracing Interface](#).



For 2D axisymmetric components an azimuthal particle velocity gives rise to a force in the radial direction, the centrifugal force. The out-of-plane velocity thus has an effect on the in-plane motion of the particles. In 2D components it is still possible to compute the out-of-plane position and velocity, but the effect on the in-plane trajectories is not as pronounced as in the axisymmetric case.

Select the **Include out-of-plane degrees of freedom** check box to define additional degrees of freedom for each particle for the out-of-plane position and velocity.



The extra degrees of freedom that are created by selecting the **Include out-of-plane degrees of freedom** check box are considered auxiliary dependent variables and are solved for using coupled first-order differential equations. If, in addition, **Newtonian** is selected from the **Formulation** list, then the resulting system of equations includes both first- and second-order equations.

Mixed first- and second-order equations systems are not supported for all solver configurations. For example, it is not possible to use explicit time stepping methods such as Dormand-Prince 5. To use explicit time-stepping methods with out-of-plane degrees of freedom, consider using the **Newtonian, first order** formulation instead.

### ADVANCED SETTINGS

These settings are the same as for [The Mathematical Particle Tracing Interface](#).

### DEPENDENT VARIABLES

The dependent variables (field variables) are the **Particle position**, **Particle position components**, **Particle velocity**, and **Particle velocity components**. The degrees of freedom for particle velocity are only used if **Newtonian, first order** is selected from the **Formulation** list. The name can be changed but the names of fields and dependent variables must be unique within a model.



- [Domain, Boundary, Pair, and Global Nodes for the Charged Particle Tracing Interface](#)
- [Theory for the Mathematical Particle Tracing Interface](#)
- [Theory for the Charged Particle Tracing Interface](#)



- With the AC/DC Module see *Magnetic Lens*: Application Library path **ACDC\_Module/Particle\_Tracing/magnetic\_lens** and *Quadrupole Mass Filter*: Application Library path **ACDC\_Module/Particle\_Tracing/quadrupole\_mass\_filter**.
- With the Plasma Module see *Ion Energy Distribution Function*: Application Library path **Plasma\_Module/Inductively\_Coupled\_Plasmas/ion\_energy\_distribution\_function**.

## *Domain, Boundary, Pair, and Global Nodes for the Charged Particle Tracing Interface*

---

The [Charged Particle Tracing Interface](#) has these domain, boundary, pair, and global nodes available from the **Physics** ribbon toolbar (Windows users), **Physics** context menu (Mac 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.

---

These nodes and subnodes are described in this section (listed in alphabetical order):

- [Attachment](#)
- [Collisions](#)
- [Current Density](#)
- [Elastic](#)
- [Electric Force](#)
- [Etch](#)
- [Excitation](#)
- [Friction Force](#)
- [Heat Source](#)
- [Ionization](#)
- [Ionization Loss](#)
- [Magnetic Force](#)
- [Nonresonant Charge Exchange](#)
- [Nuclear Stopping](#)
- [Particle Beam](#)
- [Particle-Matter Interactions](#)
- [Particle Properties](#)
- [Resonant Charge Exchange](#)
- [Space Charge Density Calculation](#)
- [Surface Charge Density](#)
- [Symmetry](#)
- [Thermionic Emission](#)
- [User Defined](#)



In the *COMSOL Multiphysics Reference Manual* see [Table 2-3](#) for links to common sections and [Table 2-4](#) 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.

---

These nodes and subnodes are described for [The Mathematical Particle Tracing Interface](#) (listed in alphabetical order):

- [Accumulator \(Boundary\)](#)
- [Accumulator \(Domain\)](#)
- [Auxiliary Dependent Variable](#)
- [Force](#)
- [Inlet](#)
- [Nonlocal Accumulator](#)
- [Outlet](#)
- [Override Properties](#)
- [Particle Continuity](#)
- [Particle Counter](#)
- [Particle-Particle Interaction](#)
- [Periodic Condition](#)
- [Release](#)
- [Release from Data File](#)
- [Release from Edge](#)
- [Release from Grid](#)
- [Release from Point](#)
- [Secondary Emission](#)
- [Velocity Reinitialization](#)
- [Wall](#) (the default boundary condition)

### *Collisions*

---

Use the **Collisions** node model the interaction between model particles and a rarefied background gas using a Monte Carlo model. Particles have a random chance to be deflected by gas molecules during each time step taken by the solver, based on the gas density and collision cross-section or collision frequency. A variety of different built-in collisions are available, each with predefined expressions for the post-collision velocity of the model particles.

By itself, the **Collisions** node does not cause any collisions to occur. It is necessary to add one or more subnodes, which correspond to different collision types. The following subnodes are available from the context menu (right-click the parent node) or from the **Physics** toolbar, **Attributes** menu: [Elastic](#), [Excitation](#), [Attachment](#), [Ionization](#), [Resonant Charge Exchange](#), [Nonresonant Charge Exchange](#), and [User Defined](#).



When the **Collisions** node and subnodes (the reaction set) are used, collisions with a background gas occur randomly and the particle velocity can change discontinuously at discrete times. The friction model, a deterministic force, can be accessed using a dedicated [Friction Force](#) node.

---



### FLUID PROPERTIES

Enter the **Background number density**  $N_d$  (SI unit:  $1/\text{m}^3$ ). The default is  $1 \times 10^{20} \text{ 1/m}^3$ .

Enter the **Background gas molar mass**  $M_g$  (SI unit: kg/mol). The default is 0.04 kg/mol.

Enter coordinates for the **Velocity field**  $\mathbf{u}$  (SI unit: m/s) based on space dimension. If another physics interface is present which computes the velocity field then this can be selected from the list.



To model the case of a stagnant background gas, leave all the velocity components as zero. If a nonzero velocity field is specified, the background gas molecules will follow a drifting Maxwellian velocity distribution.

---

Enter values or expressions for the **Temperature**  $T$  (SI unit: K) of the background gas. The default is 293.15 K.

### COLLISION STATISTICS

Select the **Count all collisions** check box to allocate an auxiliary dependent variable that records the number of times each particle collides with the background gas. If the **Collisions** node has multiple subnodes, the variable created by the **Count all collisions** check box is updated for collisions caused by any of these subnodes. In addition, the settings window for each collision type includes a **Count collisions** check box that can be used to count the number of times a specific type of collision occurs.

### NEW VALUE OF AUXILIARY DEPENDENT VARIABLES

This section is available if an [Auxiliary Dependent Variable](#) has been added to the model. The settings are the same as described for the [Wall](#) node. The auxiliary dependent variables for the particle get updated every time a collision occurs.

## AFFECTED PARTICLES

Select an option from the **Particles to affect** list to apply the force to specific particles: **All** (the default), **Single species**, or **Logical expression**. The settings are described for the [Force](#) node.



## Collisional Force Theory

---

### *Elastic*

---

The **Elastic** subnode is available from the context menu (right-click the [Collisions](#) parent node) or from the **Physics** toolbar, **Attributes** menu. Use it to model momentum-conserving collisions between model particles and background gas molecules.

## COLLISION FREQUENCY

Select an option from the **Specify** list: **Cross section** (the default) or **Collision frequency**.

- For **Collision frequency**, enter a value or expression for the **Collision frequency**  $\nu$  (SI unit: Hz). The default is  $2 \times 10^6$  Hz.
- For **Cross section** enter a value or expression for the **Collision cross section**  $\sigma$  (SI unit:  $\text{m}^2$ ). The default is  $3 \times 10^{-19}$  m.

For either choice:

- Select the **Number density specification** — **From parent** (the default) or **User defined**. For **User defined**, enter a **Background number density**  $N_d$  (SI unit:  $1/\text{m}^3$ ). The default is  $1 \times 10^{20}$   $1/\text{m}^3$ . If **From parent** is selected, the number density is inherited from the **Collisions** node.
- Select the **Molar mass specification** — **From parent** (the default) or **User defined**. For **User defined**, enter the **Background gas molar mass**  $M_g$  (SI unit: kg/mol). The default

is 0.04 kg/mol. If **From parent** is selected, the background gas molar mass is inherited from the **Collisions** node.



If the molar mass of the background gas is similar to the molar mass of the particles then the particles tend to form a Maxwellian velocity distribution function very rapidly. If the mass of the particles is very different to that of the background gas then the particles tend to have a non-Maxwellian distribution function.

If the geometry is axisymmetric and **Include out-of-plane degrees of freedom** is selected in the physics interface **Advanced Settings** section, enter a value or expression for the **Cutoff radial coordinate for out-of-plane momentum**  $r_c$  (SI unit: m). The default value is 0. If a particle undergoes a collision while its radial coordinate is less than the specified cutoff, the azimuthal component of its reinitialized velocity is set to zero. Use this setting to prevent particles from being subjected to inordinately large centrifugal forces if they undergo collisions very close to the axis of symmetry.

#### **COLLISION STATISTICS**

Select the **Count collisions** check box to count the number of times each particle collides with the background gas. The collision counter is only incremented by this instance of the **Elastic** node, not by any other collision type or by other **Elastic** nodes. To define a variable that counts all types of collisions, select the **Count all collisions** check box in the settings window for the parent **Collisions** node.

#### **NEW VALUE OF AUXILIARY DEPENDENT VARIABLES**

This section is available if an **Auxiliary Dependent Variable** has been added to the model. For each auxiliary dependent variable, select an option from the **New value of auxiliary dependent variable** list: **From parent** (the default) or **User defined**. If **From parent** is selected, the value of the auxiliary dependent variable is reinitialized according to the settings for the parent **Collisions** node.

If **User defined** is selected, select the **Assign new value to auxiliary variable** check box to assign a new value to the auxiliary dependent variable when an elastic collision occurs. Then enter the new value or expression in the field. For example, if there is an auxiliary variable,  $\psi$ , then enter a value for  $\psi_{\text{new}}$  in the field. So, to increment the value of  $\psi$  by 1 when a particle undergoes an elastic collision, enter  $\psi+1$  in the text field for  $\psi_{\text{new}}$ .

## *Excitation*

---

The **Excitation** subnode is available from the context menu (right-click the [Collisions](#) parent node) or from the **Physics** toolbar, **Attributes** menu. The **Excitation** subnode causes particles to undergo inelastic collisions with the background gas, in which some amount of kinetic energy is expended to raise the background gas molecule or ion to an excited state.

### **COLLISION FREQUENCY**

These settings are the same as for the [Elastic](#) subnode with the addition of entering a value for the **Energy loss**  $\Delta E$  (SI unit: J). The default value is 0. Typical values for electron impact collisions are between 6 eV and 10 eV.

### **COLLISION STATISTICS**

These settings are described for the [Elastic](#) node.

### **NEW VALUE OF AUXILIARY DEPENDENT VARIABLES**

These settings are described for the [Elastic](#) node.

## *Attachment*

---

The **Attachment** subnode is available from the context menu (right-click the [Collisions](#) parent node) or from the **Physics** toolbar, **Attributes** menu. The **Attachment** subnode causes particles to become attached to background gas particles, removing them from the simulation. Optionally, the attached particle can then be released. Typically, the primary particles subjected to this collision type are electrons and the attached particle is a neutralized or ionized species from the background gas.

### **COLLISION FREQUENCY**

These settings are the same as for the [Elastic](#) subnode with an additional option to select the **Release attached particle** check box. When this check box is cleared, collisions only cause the primary particle velocity and auxiliary dependent variables to be reinitialized. When this check box is selected, a secondary particle is released during each collision.

### **ION PROPERTIES**

This section is available when the **Release attached particle** check box is selected. Select an option from the **Ion properties** list when either a [Particle Properties](#) or [Override](#)

[Properties](#) node is added to the physics interface. This controls the type of species that is introduced via secondary particle emission during the attachment reaction.

#### **COLLISION STATISTICS**

These settings are described for the [Elastic](#) node.

#### **NEW VALUE OF AUXILIARY DEPENDENT VARIABLES**

These settings are described for the [Elastic](#) node.

#### **INITIAL VALUE OF AUXILIARY DEPENDENT VARIABLES**

This section is available when the **Release attached particle** check box is selected. For each auxiliary dependent variable that is defined in the model, enter the initial value of the auxiliary dependent variable for the attached particle. The default value is 0.

### *Ionization*

---

The **ionization** subnode is available from the context menu (right-click the [Collisions](#) parent node) or from the **Physics** toolbar, **Attributes** menu. The **ionization** subnode causes particles to ionize background gas molecules, releasing secondary electrons into the simulation. Optionally, the ionized background gas molecule can also be released. Typically, the primary particles subjected to this collision type are electrons and the ionized particle is taken from the background gas.

#### **COLLISION FREQUENCY**

These settings are the same as for the [Elastic](#) subnode with additional options to enter a value for the **Energy loss  $\Delta E$**  (SI unit: J) and to select the **Release ionized particle** check box. Typical energy loss values for electron impact collisions are between 12 eV and 24 eV.

#### **ELECTRON PROPERTIES**

Select an option from the **Electron properties** list, which may be any [Particle Properties](#) or [Override Properties](#) node that has been added to the physics interface.

#### **ION PROPERTIES**

This section is available when the **Release ionized particle** check box is selected. Select an option from the **Ion properties** list when either a [Particle Properties](#) or [Override Properties](#) node is added to the physics interface.

## COLLISION STATISTICS

These settings are described for the [Elastic](#) node.

## NEW VALUE OF AUXILIARY DEPENDENT VARIABLES

These settings are described for the [Elastic](#) node.

## INITIAL VALUE OF AUXILIARY DEPENDENT VARIABLES

For each auxiliary dependent variable that is defined in the model, enter the initial value of the auxiliary dependent variable for the electron and ionized particle that are released as a result of the ionization reaction. The default value is 0.

## *Resonant Charge Exchange*

---

The **Resonant Charge Exchange** subnode is available from the context menu (right-click the [Collisions](#) parent node) or from the **Physics** toolbar, **Attributes** menu. The **Resonant Charge Exchange** subnode causes particles to undergo charge exchange reactions with a background gas. Resonant charge exchange occurs between atoms of the same element or molecules of the same substance that differ in charge. The total internal energy of the particles does not change as a result of the collision. The ionized particle, neutralized particle, or both ionized and neutralized particles can be released.

## COLLISION FREQUENCY

These settings are the same as for the [Elastic](#) subnode with additional options to enter a value for the **Scattering angle in the center of mass system**  $\chi$  (SI unit: rad, default 0) and to select one of the following from the **Species to release** list — **Ion** (the default), **Neutral particle**, or **Ion and neutral particle**.

## NEUTRAL PROPERTIES

This section is available when **Neutral particle** or **Ion and neutral particle** is selected from the **Species to release** list. Select an option from the **Neutral properties** list, which may be any [Particle Properties](#) or [Override Properties](#) node that has been added to the physics interface.

## COLLISION STATISTICS

These settings are described for the [Elastic](#) node.

## NEW VALUE OF AUXILIARY DEPENDENT VARIABLES

These settings are described for the [Elastic](#) node.

## INITIAL VALUE OF AUXILIARY DEPENDENT VARIABLES

This section is available when **Neutral particle** or **Ion and neutral particle** is selected from the **Species to release** list. For each auxiliary dependent variable that is defined in the model, enter the initial value of the auxiliary dependent variable for the released neutral particle. The default value is 0.

### *Nonresonant Charge Exchange*

---

The **Nonresonant Charge Exchange** subnode is available from the context menu (right-click the [Collisions](#) parent node) or from the **Physics** toolbar, **Attributes** menu. The **Nonresonant Charge Exchange** subnode causes particles to undergo charge exchange reactions with a background gas. Unlike the [Resonant Charge Exchange](#) node, it is possible for the particle that undergoes the collision and the ionized and neutralized particles that result from it to be three different species. The ionized particle, neutralized particle, or both ionized and neutralized particles can be released.

## COLLISION FREQUENCY

These settings are the same as for the [Elastic](#) subnode with additional options to enter a value for the **Energy loss**  $\Delta E$  (SI unit: J), enter a value for the **Scattering angle in the center of mass system**  $\chi$  (SI unit: rad), and to select one of the following from the **Species to release** list: **Ion** (the default), **Neutral particle**, or **Ion and neutral particle**.

## NEUTRAL PROPERTIES

This section is available when **Neutral particle** or **Ion and neutral particle** is selected from the **Species to release** list. Select an option from the **Neutral properties** list, which may be any [Particle Properties](#) or [Override Properties](#) node that has been added to the physics interface.

## ION PROPERTIES

This section is available when **Ion** or **Ion and neutral particle** is selected from the **Species to release** list. Select an option from the **Ion properties** list, which may be any [Particle Properties](#) or [Override Properties](#) node that has been added to the physics interface.

## COLLISION STATISTICS

These settings are described for the [Elastic](#) node.

## NEW VALUE OF AUXILIARY DEPENDENT VARIABLES

These settings are described for the [Elastic](#) node.

## INITIAL VALUE OF AUXILIARY DEPENDENT VARIABLES

For each auxiliary dependent variable that is defined in the model, enter the initial value of the auxiliary dependent variable for the released neutral particle. The default value is 0.

## *User Defined*

---

The **User Defined** subnode is available from the context menu (right-click the [Collisions](#) parent node) or from the **Physics** toolbar, **Attributes** menu. The **User Defined** node allows the post-collision particle velocity to be a user-defined expression.

## COLLISION FREQUENCY

These settings are the same as for the [Elastic](#) subnode.

## REINITIALIZED PARTICLE VELOCITY

Enter values or expressions for the **Reinitialized particle velocity**  $\mathbf{v}_r$ , (SI unit: m/s) based on space dimension.

## COLLISION STATISTICS

These settings are the same as for the [Elastic](#) subnode.

## INCLUDE SECONDARY EMISSION

Select the **Include secondary emission** check box to release a secondary particle when the collision occurs.

## SECONDARY EMISSION CONDITION

This section is available when the **Include secondary emission** check box is selected.

Select a **Secondary emission condition**: **None** (the default), **Probability**, or **Expression**. For **Probability** enter the **Probability**  $\gamma$  (dimensionless) for secondary emission to occur. For **Expression** enter an **Evaluation expression**  $e$  (dimensionless) that must be nonzero for secondary emission to occur.

## SECONDARY PARTICLES

This section is available when the **Include secondary emission** check box is selected.

In the **Number of secondary particles** field  $N_s$ , specify the number of secondary particles to release per primary particle. The default value is 1.

Select an **Initial velocity** — **User defined** (the default) or **Constant speed, spherical**.



- **Constant speed, spherical** releases the secondary particles with a constant speed and spherical distribution of velocity directions. Enter a **Speed**  $v_0$  (SI unit: m/s) with which all secondary particles are released.
- **User Defined** allows for an arbitrary velocity vector to be set for the secondary particles. Enter values or expressions for the **Initial particle velocity**  $\mathbf{v}_0$  (SI unit: m/s) based on space dimension.

Select the **Offset initial position** check box to offset the positions of the secondary particles before releasing them. Then select an **Offset method** — **Using initial velocity** (the default) or **Isotropic sphere**.

- For **Using initial velocity**, enter a **Time interval**  $\Delta t$  (SI unit: s). The default is 0. Each secondary particle is then displaced by the product of its initial velocity and this time interval before being released.
- For **Isotropic sphere**, enter a **Particle displacement magnitude**  $\Delta r$  (SI unit: m). The default is 0. All secondary particles are then moved a distance equal to the displacement magnitude before they are released. If the offset would cause secondary particles to be placed outside of the modeling domain, they are instead released at the location of the primary particle.

#### INHERIT PROPERTIES

This section is available if the **Include secondary emission** check box is selected. Select an option from the **Inherit Properties** list. The **Particle Properties** node (default) is always available. If any **Override Properties** nodes have been added to the model, they can also be selected. Use this setting to determine what set of properties is given to the secondary particles that are released by this feature.

#### NEW VALUE OF AUXILIARY DEPENDENT VARIABLES

These settings are described for the **Elastic** node.

#### INITIAL VALUE OF AUXILIARY DEPENDENT VARIABLES

For each auxiliary dependent variable that is defined in the model, enter the initial value of the auxiliary dependent variable for the secondary particles that are released as a result of the **User Defined** collision. The default value is 0.

#### *Friction Force*

---

Use the **Friction Force** node to model collisions between particles and a background gas as a friction force that is proportional to the relative speed of the particles. Unlike the

collision types that are available with the [Collisions](#) node, the **Friction Force** is deterministic and does not cause the particle velocity to change discontinuously over time.

### **COLLISION FREQUENCY**

Choose an option from the **Specify** list: **Cross section and number density** or **Collision frequency**.

If **Cross section and number density** is selected, enter the following:

- **Collision cross section**  $\sigma$  (SI unit:  $\text{m}^2$ ). The default is  $3 \times 10^{-19} \text{ m}^2$ .
- **Background number density**  $N_d$  (SI unit:  $1/\text{m}^3$ ). The default is  $1 \times 10^{20} 1/\text{m}^3$ .

If **Collision frequency** is selected, enter the **Collision frequency**  $\nu$  (SI unit:  $1/\text{s}$ ). The default value is  $2 \times 10^6 \text{ Hz}$ .

### **FLUID PROPERTIES**

Enter coordinates for the **Velocity field**  $\mathbf{u}$  (SI unit:  $\text{m/s}$ ) based on space dimension.



To model the case of a stagnant background fluid, leave all the velocity components as zero.

---

### **AFFECTED PARTICLES**

Select an option from the **Particles to affect** list to apply the force to specific particles: **All** (the default), **Single species**, or **Logical expression**. The settings are described for the [Force](#) node.



[Collisional Force Theory](#)

---

### *Particle-Matter Interactions*

---

Use the **Particle-Matter Interactions** node to model the propagation of energetic ions through solid material. The [Ionization Loss](#) and [Nuclear Stopping](#) subnodes are available from the context menu (right-click the parent node) or from the **Physics** toolbar, **Attributes** menu. In order to model the interaction of particles with the target material, at least one subnode must be added.

### TARGET MATERIAL PROPERTIES

By default, the **Refractive index, real part**  $n$  (dimensionless) is taken **From material**. For **User defined** enter another value or expression.

By default, the **Density**  $\rho$  (SI unit:  $\text{kg}/\text{m}^3$ ) is taken **From material**. For **User defined** enter another value or expression.

Enter a value or expression for the **Atomic mass**  $A$  (SI unit: kg). The default value is 28.0855 amu, the relative atomic mass of silicon. Then enter a value or expression for the **Atomic number**  $Z_m$  (dimensionless). The default value is 14.

### PARTICLE PROPERTIES

Enter a value or expression for the **Atomic number**  $Z_p$  (dimensionless). The default value is 1. The particle atomic number is used to define screened Coulomb potentials for the [Nuclear Stopping](#) node.

### CUTOFF ENERGY

Select the **Stop particles with energy below cutoff** to cause particles to stop moving when their energy is less than a specified value. This check box is selected by default. Then enter a value or expression for the **Cutoff energy**  $E_c$  (SI unit: J). The default value is 8.6 eV.



In general, the **Stop particles with energy below cutoff** check box should always be selected if it is possible for ions in the material to approach extremely low energies. With the built-in ionization loss and nuclear scattering models, accurate data is not available for energies significantly less than the default value, 8.6 eV.

---

### AFFECTED PARTICLES

Use this section to determine which particles are subjected to the particle-matter interaction. By default, all particles in the model are affected. The settings are described for the [Force](#) node.

#### *Ionization Loss*

---

The **ionization Loss** subnode is available from the context menu (right-click a [Particle-Matter Interactions](#) parent node) or from the **Physics** toolbar, **Attributes** menu. Use this subnode to model the deceleration of ions by electrons in the target material.

The deceleration is treated as a continuous braking force that acts opposite the direction of motion of the particles.

#### LOSS MECHANISM

Select an option from the **Electronic Stopping Power** list: **Protons on silicon**, **Alpha particles on silicon**, or **User defined**. For **Protons on silicon** and **Alpha particles on silicon** the force that opposes the particle motion is expressed as a function of the particle kinetic energy based on empirical data tabulated in [Ref. 1](#). For **User defined** enter a value or expression for the stopping power  $S_e$  (SI unit:  $\text{m}^4/\text{s}^2$ ). The default value is  $1 \text{ MeV}\cdot\text{cm}^2/\text{g}$ .

#### *Nuclear Stopping*

---

The **Nuclear Stopping** subnode is available from the context menu (right-click a [Particle-Matter Interactions](#) parent node) or from the **Physics** toolbar, **Attributes** menu. Use this subnode to model the deflection of ions by nuclei in the target material. The deflection is modeled using a Monte Carlo model in which the particle velocity can change discontinuously in any time step.

#### LOSS MECHANISM

Select an option from the **Screening function** list: **None** (the default), **Bohr**, **Moliere**, **Lenz-Jensen**, or **Universal**. The screening function is the ratio of the electric potential of the target atom to the potential of an unscreened nucleus.

Enter a value or expression for the **Cutoff scattering angle**  $\chi_c$  (SI unit: rad). The default is 0.1 deg. Any interactions with target nuclei with a scattering angle less than the specified cutoff will be neglected. By specifying a larger cutoff scattering angle, it is possible to ignore interactions that have a small effect on particle momentum, allowing a larger time step to be taken by the solver while still accounting for the more important nuclear collisions (which usually have larger scattering angles).

#### COLLISION STATISTICS

Select the **Count collisions** check box to assign an auxiliary dependent variable for the number of collisions with the target material that each particle undergoes. The name of this collision counter is  $N_c$  preceded by the scope of the **Nuclear Stopping** node, for example `cpt.pmi1.ns1.Nc`.



### *Particle Beam*

Use the **Particle Beam** node to release a nonlaminar beam of particles normal to a boundary with a specified distribution in phase space.

The **Nonlocal Accumulator** subnode is available from the context menu (right-click the parent node) or from the **Physics** toolbar, **Attributes** menu.



For 2D axisymmetric models, always select the **Include out-of-plane degrees of freedom** check box in the physics interface **Advanced Settings** section when modeling particle beams, since the out-of-plane position and velocity are crucial in ensuring that the released beam has the specified emittance and other beam parameters.

Go to [Release](#) for information about the following sections: **Release Times**, **Release Current Magnitude**, **Inherit Properties**, **Initial Value of Auxiliary Dependent Variables**, and **Advanced Settings**.

#### **INITIAL POSITION**

Enter the **Number of particles per release**  $N$  (dimensionless). The default is 1000.

Select a **Beam position** — **From coordinates** (the default), **Centroid of selection**, or **Selected point**. This option determines how the position of the center of the beam is computed. For 2D axisymmetric model components, an additional option is available — **Axis of symmetry** (the default).

For **From coordinates** enter coordinates for the **Beam center location**  $\mathbf{r}_c$  (SI unit: m). When **Selected point** is selected, the **Reference Point Selection** section is shown. Add at least one point to the selection to specify the center of the beam.

## INITIAL TRANSVERSE VELOCITY

The **Equation** section of the Particle Beam feature contains images indicating how the phase space ellipses are constructed for the different settings available. To enable this, use **Show>Equation Sections** in the **Model Builder** toolbar.

- Select a **Sampling from phase space ellipse: KV, Waterbag, Parabolic** or **Gaussian** (the default). The **KV** distribution places particles uniformly on the surface of a hyperellipsoid in 4-dimensional phase space. In 3D, this results in a particle distribution which is uniform in physical space, i.e. the distance between particles is roughly the same throughout the beam cross-section. The **Waterbag** distribution places particles uniformly inside the volume of a hyperellipsoid in 4-dimensional phase space. This results in a distribution in physical space in which particles are placed preferentially towards the center of the beam. The **Parabolic** distribution is similar to the **Waterbag** distribution, but more particles are placed towards the center of the beam in physical space. The **Gaussian** distribution fills the volume of a hyperellipsoid with a normal distribution density profile in 4-dimensional phase space. Similar to the **Waterbag** and **Parabolic** distributions, particles are preferentially placed closer to the center of the beam, although in this case, some particles can be placed a significant distance from the center
- In 3D, select a **Beam symmetry: Symmetric** (the default) or **Asymmetric**. If **Asymmetric** is selected, two different sets of beam parameters, such as emittance and Twiss parameters, can be specified for the two transverse directions. If **Symmetric** is selected, the same beam parameters are used for both transverse directions.
- Select a **Beam orientation: Upright** (the default) or **Not upright**. **Upright** means that the semimajor and semiminor axes of the phase space ellipses are parallel to the phase space coordinate axis. For a beam that is not subjected to any forces, **Upright** means that the beam is released at the location of the beam waist. **Not upright** allows the phase space ellipse to be rotated about its centroid.
- Select a **Transverse velocity distribution specification: Specify emittance and Twiss parameters** (the default) or **Specify phase space ellipse dimensions**. Twiss parameters are generally favored by accelerator physicists, as they provide a standard method of defining the geometric properties of the phase space ellipse. **Specifying phase space ellipse dimensions** is often favored by spectrometer designers, as it allows the dimensions of the phase space ellipse and rotation angle to be specified directly.
- In some cases it is possible to specify the **Brightness  $B$**  (SI unit:  $A/m^2$ ) instead of emittance. The default value is  $1 \text{ mA}/m^2$ . The beam emittance is then computed from the brightness and the specified **Release current magnitude**. It is possible to

enter a value or expression for the brightness of the beam if the following conditions are met:

- The model component is 3D or 2D axisymmetric.
  - **Specify current** is selected from the **Particle release specification** list in the physics interface **Particle Release and Propagation** section.
  - **Symmetric** is selected from the **Beam symmetry** list.
  - **Specify emittance and Twiss parameters** is selected from the **Transverse velocity distribution specification** list.
  - **Specify brightness** is selected from the **Emittance specification** list, which is only shown if the preceding conditions are met.
- For 3D components, when the **Beam symmetry** is set to **Asymmetric**, it is possible to specify different phase space distributions in two orthogonal transverse directions. To decide how these transverse directions are oriented, select a **Transverse direction specification: User defined** or **Parallel to reference edge**. When **Parallel to reference edge** is selected, the **Reference Edge Selection** section is shown. Add an edge to this selection to define the coordinate system with respect to which the beam emittance and Twiss parameters are defined. Quantities with subscript 1 ( $\epsilon_{\text{rms},1}$ ,  $\beta_1$ , etc.) are used to determine the distributions of the position and velocity components parallel to the selected edge. Quantities with subscript 2 correspond to the position and velocity components perpendicular to the selected edge and to the direction of beam propagation. If **User defined** is selected, enter components of the **Transverse beam direction  $t_1$**  (dimensionless) directly. The default is the positive  $y$ -direction.

Depending on the options selected above, it is possible to enter values for different physical quantities which describe the phase space ellipse dimensions. These are summarized below and in the following table:

- The **Twiss parameter beta**  $\beta$  (SI unit: m). The default value is 1 m. The product of this and the emittance give an indication of the transverse size of the particle beam. For a given value of the beam emittance, increasing  $\beta$  causes the particle distribution in transverse position space to be broadened and the distribution in transverse velocity space to be tightened.
- The **I-RMS beam emittance**  $\epsilon_{\text{rms}}$  (SI unit: m). The default is 1 mm. The emittance is a measure of the average spread of the particles in phase space. Qualitatively, a larger emittance means that the area occupied by the distribution of particles in phase space is larger.

- The **Brightness**  $B$  (SI unit:  $A/m^2$ ). The default is  $1 \text{ mA}/m^2$ . For symmetric beams the brightness can be used instead of specifying the emittance. This option is only available when a certain combination of settings are applied, as explained in the previous list.
- The **Twiss parameter alpha**  $\alpha$  (dimensionless). The default is 0. When this value is non-zero, the phase space ellipse becomes rotated about the position — velocity axis. When **Upright** is selected from the **Beam orientation** list,  $\alpha = 0$ . The remaining Twiss parameter  $\gamma$  (SI unit:  $1/m$ ) is not specified because it is derived from  $\alpha$  and  $\beta$ .
- The **Maximum transverse displacement**  $x_m$  (SI unit: m). The default is 0.1 m. Typically, this value should be the same as the initial beam radius.
- The **Maximum relative transverse velocity**  $x_m'$  (dimensionless). The default is 0.04. Larger values indicate that the particles will have higher velocities in the transverse direction with respect to the longitudinal direction.
- The **Rotation angle**  $\theta$  (SI unit: rad). The default is 0. This rotates the phase space ellipse about the position-velocity axis.

Two separate values for all of the above options can be entered in 3D when the **Beam symmetry** is set to **Asymmetric**.



TABLE 4-1: SUMMARY OF INITIAL TRANSVERSE VELOCITY OPTIONS

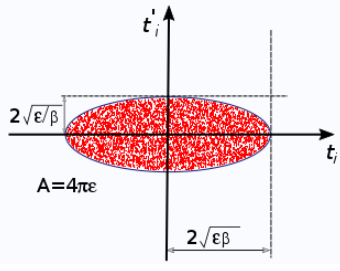
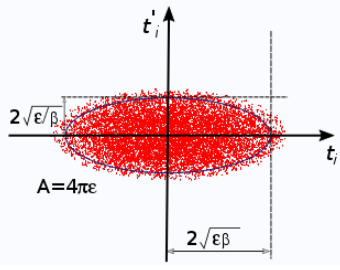
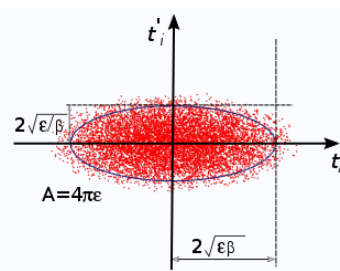
SAMPLING	ORIENTATION	TRANSVERSE VELOCITY SPECIFICATION	IMAGE	AVAILABLE INPUT
KV	Upright	Twiss		$\beta, \epsilon_{\text{rms}}$ $\beta, B$
Waterbag	Upright	Twiss		$\beta, \epsilon_{\text{rms}}$ $\beta, B$
Parabolic	Upright	Twiss		$\beta, \epsilon_{\text{rms}}$ $\beta, B$

TABLE 4-1: SUMMARY OF INITIAL TRANSVERSE VELOCITY OPTIONS

SAMPLING	ORIENTATION	TRANSVERSE VELOCITY SPECIFICATION	IMAGE	AVAILABLE INPUT
Gaussian	Upright	Twiss		$\beta, \epsilon_{\text{rms}}$ $\beta, B$
KV	Not upright	Twiss		$\beta, \epsilon_{\text{rms}}, \alpha$ $\beta, B, \alpha$
Waterbag	Not upright	Twiss		$\beta, \epsilon_{\text{rms}}, \alpha$ $\beta, B, \alpha$

TABLE 4-1: SUMMARY OF INITIAL TRANSVERSE VELOCITY OPTIONS

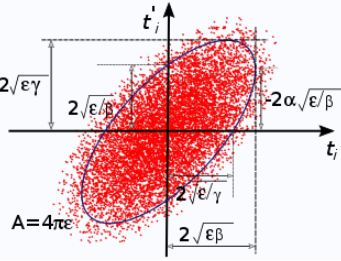
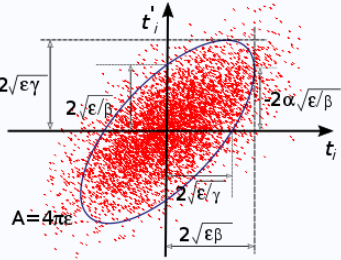
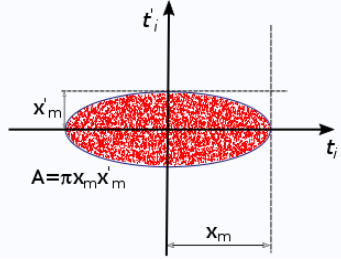
SAMPLING	ORIENTATION	TRANSVERSE VELOCITY SPECIFICATION	IMAGE	AVAILABLE INPUT
Parabolic	Not upright	Twiss		$\beta, \epsilon_{\text{rms}}, \alpha$ $\beta, B, \alpha$
Gaussian	Not upright	Twiss		$\beta, \epsilon_{\text{rms}}, \alpha$ $\beta, B, \alpha$
KV	Upright	Dimensions		$x_m, x_m'$

TABLE 4-1: SUMMARY OF INITIAL TRANSVERSE VELOCITY OPTIONS

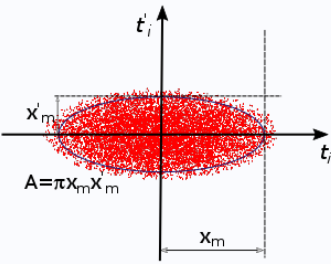
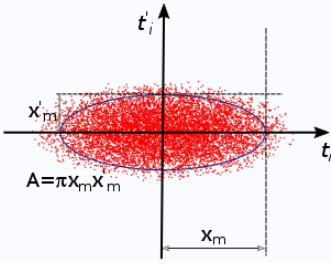
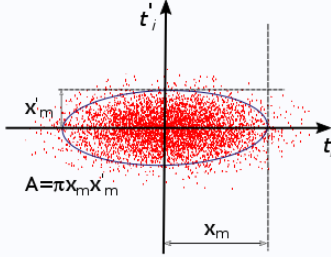
SAMPLING	ORIENTATION	TRANSVERSE VELOCITY SPECIFICATION	IMAGE	AVAILABLE INPUT
Waterbag	Upright	Dimensions	 <p>The diagram shows a 2D coordinate system with a horizontal axis labeled <math>t_i</math> and a vertical axis labeled <math>t'_i</math>. A dense cloud of red points is centered at the origin, forming a horizontal ellipse. A horizontal dashed line extends from the center to the right edge of the ellipse, labeled <math>x'_m</math>. A vertical double-headed arrow below the ellipse indicates its width, labeled <math>x_m</math>. The formula <math>A = \pi x_m x'_m</math> is written in the lower-left quadrant of the plot area.</p>	$x_m, x'_m$
Parabolic	Upright	Dimensions	 <p>The diagram shows a 2D coordinate system with a horizontal axis labeled <math>t_i</math> and a vertical axis labeled <math>t'_i</math>. A dense cloud of red points is centered at the origin, forming a horizontal ellipse. A horizontal dashed line extends from the center to the right edge of the ellipse, labeled <math>x'_m</math>. A vertical double-headed arrow below the ellipse indicates its width, labeled <math>x_m</math>. The formula <math>A = \pi x_m x'_m</math> is written in the lower-left quadrant of the plot area.</p>	$x_m, x'_m$
Gaussian	Upright	Dimensions	 <p>The diagram shows a 2D coordinate system with a horizontal axis labeled <math>t_i</math> and a vertical axis labeled <math>t'_i</math>. A dense cloud of red points is centered at the origin, forming a horizontal ellipse. A horizontal dashed line extends from the center to the right edge of the ellipse, labeled <math>x'_m</math>. A vertical double-headed arrow below the ellipse indicates its width, labeled <math>x_m</math>. The formula <math>A = \pi x_m x'_m</math> is written in the lower-left quadrant of the plot area.</p>	$x_m, x'_m$

TABLE 4-1: SUMMARY OF INITIAL TRANSVERSE VELOCITY OPTIONS

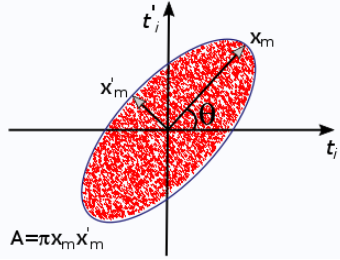
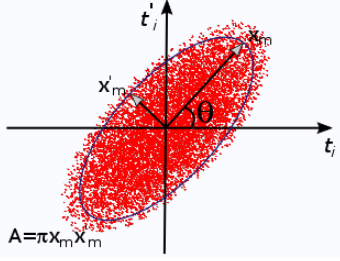
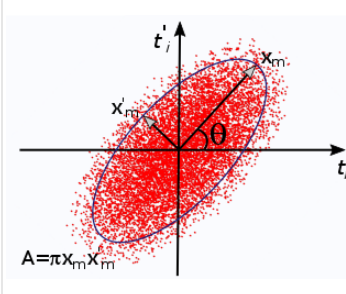
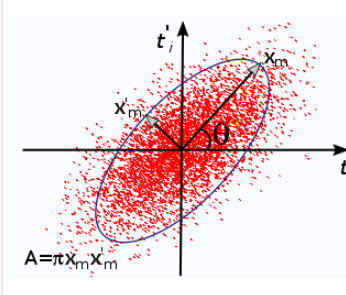
SAMPLING	ORIENTATION	TRANSVERSE VELOCITY SPECIFICATION	IMAGE	AVAILABLE INPUT
KV	Not upright	Dimensions		$x_m, x'_m, \theta$
Waterbag	Not upright	Dimensions		$x_m, x'_m, \theta$

TABLE 4-1: SUMMARY OF INITIAL TRANSVERSE VELOCITY OPTIONS

SAMPLING	ORIENTATION	TRANSVERSE VELOCITY SPECIFICATION	IMAGE	AVAILABLE INPUT
Parabolic	Not upright	Dimensions		$x_m, x_m', \theta$
Gaussian	Not upright	Dimensions		$x_m, x_m', \theta$

### INITIAL LONGITUDINAL VELOCITY

Select a **Longitudinal velocity specification**: **Specify kinetic energy** (the default) or **Specify velocity**. This determines the physical quantity of the remaining inputs in this section, which are used to initialize the longitudinal component of the particle velocity. Select a **Longitudinal velocity distribution**: **None** (the default), **Normal**, **Uniform** or **List of values**. The following options are available.

- For **None**, enter either a **Kinetic energy**  $E$  (SI unit: J), default 1 eV or a **Velocity magnitude**  $V$  (SI unit: m/s), default 1 m/s. This option releases one particle for each initial position.
- For **Normal**, enter either a **Mean kinetic energy**  $E_m$  (SI unit: J), default 1 eV and a **Kinetic energy standard deviation**  $\sigma$  (SI unit: J), default 0.1 eV, or a **Mean velocity magnitude**  $V_m$  (SI unit: m/s), default 1 m/s and **Velocity magnitude standard deviation**  $\sigma$  (SI unit: m/s), default 0.1 m/s. In both cases, also enter the **Number of velocity values**. The default is 1, and this option determines the number of particles released for each initial position, meaning this value multiplies the total number of

particles released. The longitudinal component of the velocity is sampled from a normal distribution, meaning that roughly 68% of the particles will have an initial velocity within one standard deviation, and 95% within two standard deviations.

- For **Uniform**, enter either a **Minimum kinetic energy**  $E_{\min}$  (SI unit: J), default 1 eV and a **Maximum kinetic energy**  $E_{\max}$  (SI unit: J), default 2 eV, or a **Minimum velocity magnitude**  $V_{\min}$  (SI unit: m/s), default 1 m/s and **Maximum velocity magnitude**  $V_{\max}$  (SI unit: m/s), default 2 m/s. In both cases, also enter the **Number of velocity values**. The default is 1, and this option determines the number of particles released for each initial position, meaning this value multiplies the total number of particles released. The uniform distribution will always include the minimum and maximum velocity or energy values if the **Number of velocity values** is greater than 1. If the **Number of velocity values** is 1, the velocity magnitude is simply the mean of the minimum and maximum values.
- For **List of values**, enter either a list of **Kinetic energy values** (SI unit: J) or a list of **Velocity values** (SI unit: m/s). The length of this list determines how many particles are released per initial position.

Select the **Reverse direction** check box to reverse the longitudinal direction of the released particles. This is useful when the feature selection is an interior boundary, and the normal direction is ambiguous. The direction of beam propagation is indicated by arrows on the selected boundaries in the Graphics window.



## Particle Beam Theory

---

### *Thermionic Emission*

---



The **Thermionic Emission** node is only available in 3D.

It is also only available when **Newtonian** or **Newtonian, first order** is selected from the **Formulation** list and **Specify current** is selected from the **Particle release specification** list in the physics interface **Particle Release and Propagation** section.

Use the **Thermionic Emission** node to model the release of electrons from a hot cathode. The properties specified in the [Particle Properties](#) node should be appropriate for electrons, that is, a mass of `me_const` and a charge number of -1.

Go to [Release](#) for information about the following sections: **Inherit Properties**, **Initial Value of Auxiliary Dependent Variables**, and **Advanced Settings**.

### **SURFACE PROPERTIES**

Enter a value or expressions for the **Temperature**  $T$  (SI unit: K) of the cathode. If the temperature is computed by another physics interface then it can be selected from the list.

Enter a value or expression for the **Metal work function**  $\Phi$  (SI unit: V). The default expression is 4.5[V].

Enter a value or expression for the **Effective Richardson constant**  $A^*$  (SI unit:  $A/(m^2 K^2)$ ). The default expression is  $110[A/(K \cdot cm)^2]$ .

Enter a value for the **Number of particles per release**  $N$  (dimensionless). The default is 100.

### **INITIAL VELOCITY**

Enter the **Number of particles in velocity space**  $N_{vel}$  (dimensionless). The default is 200. This determines the number of particle velocity vectors to sample at each release point. Therefore the total number of particles released by the feature is usually  $N \times N_{vel}$ , but may be greater if any auxiliary dependent variables are also sampled from distributions.

The thermionic emission of electrons from the surface follows an axially symmetric probability distribution function of particle velocities that is centered about the surface normal. Select the **Reverse direction** check box to reverse the direction of the surface normal for the purpose of initializing the particle velocities. This is useful when releasing particles from internal boundaries where the default surface normal might not point into the desired adjacent domain. The surface normals on selected boundaries are indicated by arrows in the Graphics window.

Select an option from the **Weighting of macroparticles** list: **Uniform current** (the default), **Uniform speed intervals**, or **Uniform energy intervals**. Each model particle released by the **Thermionic emission** feature is a macroparticle representing a certain number of emitted electrons per unit time.

When **Uniform current** is selected, each model particle represents the same number of electrons and the initial particle speeds are sampled from a probability distribution function based on a population of electrons following a Maxwell-Boltzmann distribution.



When **Uniform speed intervals** is selected, particles are sampled from uniform intervals from 0 to a maximum speed, which is expressed in terms of the average thermal energy of the electrons. Specify the **Maximum multiple of thermal energy  $n$**  (default 10).

When **Uniform energy intervals** is selected, particles are sampled from uniform intervals from 0 to a maximum energy, which is a multiple of the average thermal energy of the electrons. Specify the **Maximum multiple of thermal energy  $n$**  (default 10).

For **Uniform speed intervals** or **Uniform energy intervals** each model particle can be weighted differently so that it represents a unique number of electrons per unit time. Particles that are initialized with a more probable initial speed or energy are assigned a greater weight for the purpose of computing charge density on domains and current density at boundaries.



Thermionic Emission Theory

---



*Planar Diode*: Application Library path

**Particle\_Tracing\_Module/Charged\_Particle\_Tracing/planar\_diode**

---

### *Particle Properties*

---

Use the **Particle Properties** node to specify the particle mass and charge number when using the **Newtonian** or **Newtonian, first order** formulation. Specify the particle velocity when using the Massless formulation.

#### **PARTICLE MASS**

This section is shown when **Newtonian** or **Newtonian, first order** is selected as the **Formulation**.

Enter a value or expression for the **Particle mass  $m_p$**  (SI unit: kg). The default expression is `me_const`, which is a predefined constant within COMSOL Multiphysics for the electron mass,  $9.10938188 \times 10^{-31}$  kg.

If the **Relativistic correction** check box is selected in the physics interface **Advanced Settings** section, instead specify the **Particle rest mass  $m_r$**  (SI unit: kg). The default expression is `me_const`.

## CHARGE NUMBER

This section is shown when **Newtonian** or **Newtonian, first order** is selected as the **Formulation**.

Enter a value for the **Charge number  $Z$**  (dimensionless). The default is  $-1$  corresponding to the charge number of an electron.

## PARTICLE VELOCITY

This section is shown when **Massless** is selected as the **Formulation**.

Enter a vector for the **Particle velocity  $\mathbf{v}$**  (SI unit: m/s) based on space dimension. The **Massless** formulation means that the particles follow streamlines of the particle velocity expression.

## *Symmetry*

---

Use the **Symmetry** node to define a plane of symmetry at selected boundaries in the geometry.

For particle tracing models, the physical interpretation of plane symmetry is as follows: for every particle that exits the domain through a boundary assigned a **Symmetry** node, a new particle enters the modeling domain at the same location at the same time, with an incoming velocity that mirrors the outgoing velocity of the exiting particle. Thus the model particle is specularly reflected as if it hit a **Wall** with the **Bounce** condition.

## *Electric Force*

---



The **Electric Force** node is also only available when **Newtonian** or **Newtonian, first order** is selected as the **Formulation** in the physics interface **Particle Release and Propagation** section.

---

Use the **Electric Force** node to define the electric part of the Lorentz force. The particles are accelerated in the parallel or antiparallel to the electric field depending on their charge. The force is specified via an electric potential or the electric field. For cases where the field was computed in the frequency domain, the force can be computed by multiplying the field by the phase angle. Additionally, piecewise polynomial recovery can be used, which can give a more accurate representation of the specified electric field.

## ELECTRIC FORCE

Select an option from the **Specify force using** list — **Electric field** (the default) or **Electric potential**.

- For **Electric field** enter values or expressions in the table for the **Electric field**  $\mathbf{E}$  (SI unit: V/m) based on space dimension. If the electric field is computed by another physics interface then it can be selected from the list.
- For **Electric potential** enter a value or expression for **Electric potential**  $V$  (SI unit: V). If the electric potential is computed by another physics interface then it can be selected from the list.

## ADVANCED SETTINGS

Select the **Multiply force by phase angle** check box in cases where the field is computed in the frequency domain. Then set the **Initial phase angle**  $\phi_0$  (SI unit: rad). The default value is 0. When the particle trajectories are computed, the value of the field used to compute the force is computed from the complex field via:

$$\mathbf{E} = \text{real}(\tilde{\mathbf{E}} \exp(j(\omega t + \phi_0)))$$

where  $\mathbf{E}$  is the electric field value used to compute the force,  $\tilde{\mathbf{E}}$  (SI unit: V/m) is the complex-valued electric field, and  $\omega$  (SI unit: rad/s) is the angular frequency. If the field is computed using a different study type, such as a Stationary or Time Dependent study, this option has no effect.

Select the **Use piecewise polynomial recovery on field** check box to smooth the electric field using piecewise polynomial recovery. This can give a much more accurate representation of the electric field as it uses information on adjacent mesh elements to reconstruct the field. If a coarse mesh is used to compute the field then this option can be especially useful.

Use the **Particles to affect** list to apply the force to specific particles. The available settings are the same as for the [Force](#) node.





The **Magnetic Force** node is only available when **Newtonian** or **Newtonian, first order** is selected as the **Formulation** in the physics interface **Particle Release and Propagation** section

---

Use the **Magnetic Force** node to define the magnetic component of the Lorentz force. This causes the particle to curve perpendicularly to the particle velocity and magnetic field. A magnetic force alone does no work on the particles, so in the absence of any other external forces, the particle retains its original energy. Some small variation in the particle kinetic energy can be expected, but it is just due to numerical error and can usually be reduced by adjusting the solver settings. The force is specified via a magnetic flux density.

#### **MAGNETIC FORCE**

Enter values or expressions in the table for the **Magnetic flux density  $\mathbf{B}$**  (SI unit: T) based on space dimension, or, if the magnetic flux density is computed by another physics interface, select another option from the list.

##### *Computing a Force due to Earth's Magnetic Field*

Select **Earth's magnetic field** from the **Magnetic flux density** list to use measurements of the Earth's magnetic field to compute the magnetic force. The external data is from the International Geomagnetic Reference Field (IGRF; [Ref. 2](#)). Select an option from the **Location specification** list: **From particle positions** (default) or **Geographic location**. Enter a value or expression for the **Epoch  $E$**  (dimensionless) to determine which measurements are used to compute the Earth's magnetic field. The default value is 2015. The IGRF data is tabulated in 5-year increments, and data from 1950 to 2015 is included for the **Magnetic Force** node.

If **From particle positions** is selected, the magnetic force is computed by treating the particle coordinates as position vector components with respect to the center of the Earth, such that the geographic North pole points in the positive  $z$ -direction and the Prime Meridian intersects the positive  $x$ -axis.

If **Geographic location** is selected the magnetic field components are assigned constant values based on the specified location. This is appropriate when the modeling domain is very small relative to the length scales over which the Earth's magnetic field changes significantly. By default, the magnetic field is determined based on the assumption that the positive  $x$ -axis points to the North, the positive  $y$ -axis points to the West, and the

positive  $z$ -axis points upward. Select an option from the **Location defined by** list: **City** (the default) or **Coordinates**. If **City** is selected, choose one of several major cities available in the **City** list. The default is **Las Vegas, USA**. If **Coordinates** is selected, enter the **Latitude**  $\Phi$  and **Longitude**  $\Theta$ . Positive values indicate latitude North of the equator and longitude East of the Prime Meridian, respectively. The default values are 36.1°N, 115.2°W.



For the **Latitude** and **Longitude** it is important to indicate that the entered quantities are in degrees, usually by including [deg] after the numeric value. Otherwise, numeric values with no specified unit will be interpreted in the default plane angle unit, which is typically the radian.

For both options in the **Location defined by** list, enter the **Altitude**  $h$  (SI unit: m). The default is 0.

#### ADVANCED SETTINGS

Select the **Multiply force by phase angle** check box in cases where the field is computed in the frequency domain. Then set the **Initial phase angle**  $\phi_0$  (SI unit: rad). The default value is 0 radians. When the particle trajectories are computed, the value of the field used to compute the force is computed from the complex field via:

$$\mathbf{B} = \text{real}(\tilde{\mathbf{B}} \exp(j(\omega t + \phi_0)))$$

where  $\mathbf{B}$  is the magnetic flux density value used to compute the force,  $\tilde{\mathbf{B}}$  is the complex-valued magnetic flux density, and  $\omega$  is the angular frequency. If the field is computed using a different study type, such as a Stationary or Time Dependent study, this option has no effect.

Select the **Use piecewise polynomial recovery on field** check box to smooth the magnetic flux density using piecewise polynomial recovery. This can give a much more accurate representation of the magnetic flux density as it uses information on adjacent mesh elements to reconstruct the field. If a coarse mesh is used to compute the field then this option can be especially useful.

Use the **Particles to affect** list to apply the force to specific particles. The available settings are the same as for the [Force](#) node.



Magnetic Force Theory

## Space Charge Density Calculation

---

Use the **Space Charge Density Calculation** node if space charge effects are important. This node defines a dependent variable for the space charge density in the selected domains. Optionally, the current density can also be computed.



The **Space Charge Density Calculation** node computes space charge density and current density in the same way as the [Electric Particle Field Interaction](#) and [Magnetic Particle Field Interaction](#) Multiphysics nodes, respectively, but does not automatically include the accumulated variables as source terms when computing the fields. For modeling bidirectionally coupled particle-field interactions, consider using [The Particle Field Interaction, Non-Relativistic Interface](#) or [The Particle Field Interaction, Relativistic Interface](#) instead of the **Space Charge Density Calculation** node.

For more information about the calculation of the space charge density, see [Space Charge Density Calculation](#) in the section [Theory for the Particle Field Interaction, Non-Relativistic Interface](#). For more information about the calculation of current density, see [Current Density Calculation](#) in the section [Theory for the Magnetic Particle Field Interaction, Relativistic Interface](#).

---

### SPACE CHARGE DENSITY CALCULATION

If **Specify release times** is selected from the **Particle release specification** list in the physics interface **Particle Release and Propagation** section, enter a value or expression for the **Charge multiplication factor**  $n$  (dimensionless). The default is  $10^6$ . The **Charge multiplication factor** can be used to represent each particle as a group of  $n$  particles that follow the same trajectory. This means that the magnitude of the contribution to the space charge density by each particle is multiplied by  $n$ . It is thus possible to model space charge effects without allocating degrees of freedom for an overwhelming number of particles.



If **Specify current** is selected from the **Particle release specification** list in the physics interface **Particle Release and Propagation** section, the **Charge multiplication factor**  $n$  cannot be specified because the number of charged particles represented by each model particle is instead controlled by the **Release current magnitude**, which is specified in the settings for release features such as the [Release](#) and [Inlet](#) nodes.

Select the **Compute current density** check box to allocate additional degrees of freedom for the components of the current density.



The following features are specific applications of the core functionality used in the [Accumulator \(Boundary\)](#) subnode, described in [The Mathematical Particle Tracing Interface](#).

---

### *Surface Charge Density*

---

When the **Particle release specification** is set to **Specify release times**, the **Surface Charge Density** subnode is available from the context menu (right-click the [Wall](#) parent node) or from the **Physics** toolbar, **Attributes** menu. The **Surface Charge Density** subnode defines an accumulated variable on each boundary element in the selection of the parent node that calculates the charge density of deposited particles.

---



[Surface Charge Density Theory](#)

---

### *Current Density*

---

When the **Particle release specification** is set to **Specify current**, the **Current Density** subnode is available from the context menu (right-click the [Wall](#) parent node) or from the **Physics** toolbar, **Attributes** menu. The **Current Density** subnode uses accumulated variables on each boundary element in the selection of the parent node to calculate the current density due to the impact of charged particles.

#### **CURRENT DENSITY**

Select a **Type**: **Current density** or **Normal current density**. For **Current density** accumulated variables are defined for the components of the current density vector. For **Normal current density** one accumulated variable is defined for the normal current density in each boundary element.

---



[Current Density Theory](#)

---

## Heat Source

---

When the **Particle release specification** is set to **Specify current**, the **Heat Source** subnode is available from the context menu (right-click the **Wall** parent node) or from the **Physics** toolbar, **Attributes** menu. The **Heat Source** subnode defines an accumulated variable on the selection of the parent node that calculates a boundary heat source using the kinetic energy of the incident particles.

---



### Heat Source Theory

---

## Etch

---

When the **Particle release specification** is **Specify current**, the **Etch** subnode is available from the context menu (right-click the **Wall** parent node) or from the **Physics** toolbar, **Attributes** menu, which computes the etch rate due to the impact of particles.

### ETCH

Select an **Angular dependence model**: **Argon on polysilicon** (the default) or **Expression**. **Argon on polysilicon** defines the dependence of the etch rate on the angle of incidence by using an empirical model for the impact of argon ions on polysilicon. For **Expression** enter an **Angular dependence function**  $Y$  (dimensionless). The default is 1.

Enter a **Threshold energy**  $E_{th}$  (SI unit: J). The default is 50[eV].

Enter a **Slope of etch yield curve**  $Y$  (SI unit: J). The default is 625[eV].

Enter a **Plasma type**: **Collisionless** (default) or **Collisional**. For **Collisional** enter a value or expression for the **Normal ion current density**  $\mathbf{n} \cdot \mathbf{J}_i$  (SI unit: A/m<sup>2</sup>). The default is 0. If the normal ion current density is computed by another physics interface then it can be selected from the list.

### SURFACE PROPERTIES

The following settings are required if the **Angular dependence model** is set to **Expression**.

Enter the **Molecular weight of surface material**  $M_s$  (SI unit: kg/mol). The default is 0.028[kg/mol].

Enter the **Mass density of surface**  $\rho$  (SI unit: kg/m<sup>3</sup>). The default is 2320[kg/m<sup>3</sup>].





Etch Theory

---

# Theory for the Charged Particle Tracing Interface

The Charged Particle Tracing Interface theory is described in this section:

- [Introduction to the Charged Particle Tracing Interface Theory](#)
- [Electric Force Theory](#)
- [Magnetic Force Theory](#)
- [Collisional Force Theory](#)

## *Introduction to the Charged Particle Tracing Interface Theory*

---

Motion of charged particles in electromagnetic field is best understood by starting with the Lagrangian for a charge in an electromagnetic field:

$$L = -m_p c^2 \sqrt{1 - \mathbf{v} \cdot \mathbf{v} / c^2} + Ze\mathbf{A} \cdot \mathbf{v} - ZeV$$

where

- $m_p$  (SI unit: kg) is the particle mass,
- $c = 2.99792458 \times 10^8$  m/s is the speed of light in a vacuum,
- $\mathbf{v}$  (SI unit: m/s) is the particle velocity,
- $Z$  (dimensionless) is the particle charge,
- $e = 1.602176565 \times 10^{-19}$  C is the elementary charge,
- $\mathbf{A}$  (SI unit: Wb/m) is the magnetic vector potential, and
- $V$  (SI unit: V) is the electric scalar potential.

For low velocities, after subtracting the rest energy, the Lagrangian becomes:

$$L = \frac{m_p v^2}{2} + Ze\mathbf{A} \cdot \mathbf{v} - ZeV$$

The equations of motion are given by the Lagrange equation:

$$\frac{d}{dt} \left( \frac{\partial L}{\partial \mathbf{v}} \right) = \frac{\partial L}{\partial \mathbf{q}}$$

where  $\mathbf{q}$  (SI unit: m) is the particle position vector. The right-hand side of the Lagrange equation is:

$$\frac{\partial L}{\partial \mathbf{q}} = \nabla L = Ze(\nabla(\mathbf{A} \cdot \mathbf{v}) - \nabla V)$$

Using some elementary vector calculus, this can be rewritten as:

$$\frac{\partial L}{\partial \mathbf{q}} = Ze(\mathbf{v} \cdot \nabla)\mathbf{A} + Ze(\mathbf{v} \times \nabla \times \mathbf{A}) - Ze\nabla V$$

So, the Lagrange equation becomes:

$$\frac{d}{dt}(m_p \mathbf{v} + Ze\mathbf{A}) = Ze(\mathbf{v} \cdot \nabla)\mathbf{A} + Ze(\mathbf{v} \times \nabla \times \mathbf{A}) - Ze\nabla V \quad (4-1)$$

The second term on the left-hand side of [Equation 4-1](#) represents the total differential which can be expressed as:

$$\frac{d}{dt}(Ze\mathbf{A}) = Ze\left(\frac{\partial \mathbf{A}}{\partial t} + (\mathbf{v} \cdot \nabla)\mathbf{A}\right) \quad (4-2)$$

Inserting [Equation 4-2](#) into [Equation 4-1](#) results in:

$$\frac{d}{dt}(m_p \mathbf{v}) = -Ze\frac{\partial \mathbf{A}}{\partial t} - Ze\nabla V + Ze(\mathbf{v} \times \nabla \times \mathbf{A}) \quad (4-3)$$

Defining the electric field as:

$$\mathbf{E} = -\frac{\partial \mathbf{A}}{\partial t} - \nabla V$$

and the magnetic flux density as:

$$\mathbf{B} = \nabla \times \mathbf{A}$$

the equation of motion for a charged particle in an electromagnetic field can be written as:

$$\frac{d}{dt}(m_p \mathbf{v}) = -Ze\mathbf{E} + Ze(\mathbf{v} \times \mathbf{B}) \quad (4-4)$$

The term on the right-hand side of [Equation 4-4](#) is called the Lorentz force. So far, only the [Electric Force](#) and [Magnetic Force](#) have been considered. Additional forces, such those introduced by the [Friction Force](#) and [Ionization Loss](#) features, can be added

to the right-hand side if necessary.

### *Electric Force Theory*

---

The [Electric Force](#) feature adds the following contribution to the total force  $\mathbf{F}_t$ :

$$\mathbf{F}_t = Ze\mathbf{E}$$

where  $\mathbf{E}$  (SI unit: V/m) is the electric field. When an electric potential  $V$  (SI unit: V) is used to specify the electric field, the following is used:

$$\mathbf{E} = -\nabla V$$

When the electric field is computed in the frequency domain it is complex valued. The field must be cast into a real value which depends on the angular frequency and the simulation time:

$$\mathbf{E}(t) = \text{real}(\tilde{\mathbf{E}}\exp(j(\omega t + \phi_0)))$$

where  $\tilde{\mathbf{E}}$  is the complex-valued electric field,  $\omega$  is the angular frequency,  $\phi_0$  is the initial phase angle, and  $t$  is time.

### *Magnetic Force Theory*

---

The [Magnetic Force](#) feature adds the following contribution to the total force  $\mathbf{F}_t$ :

$$\mathbf{F}_t = Ze(\mathbf{v} \times \mathbf{B})$$

where  $\mathbf{B}$  (SI unit: T) is the magnetic flux density. When the magnetic flux density is computed in the frequency domain it is complex valued. The field must be cast into a real value which depends on the angular frequency and the simulation time:

$$\mathbf{B}(t) = \text{real}(\tilde{\mathbf{B}}\exp(j(\omega t + \phi_0)))$$

where  $\tilde{\mathbf{B}}$  is the complex-valued electric field,  $\omega$  is the angular frequency,  $\phi_0$  is the initial phase angle, and  $t$  is time.

#### *Computing Earth's Magnetic Field*

It is possible to exert a magnetic force on particles based on the Earth's magnetic field. The data for Earth's magnetic field is the International Geomagnetic Reference Field (IGRF; [Ref. 2](#)) and is based on the Geomag 7.0 software. The IGRF data consists of

sets of Gauss coefficients that define a spherical harmonic expansion for the magnetic scalar potential:

$$V(r, \Phi, \Theta) = R_e \sum_{l=1}^L \sum_{m=-l}^l \left(\frac{R_e}{r}\right)^{l-1} (g_l^m \cos(m\Phi) + h_l^m \sin(m\Phi)) P_l^m \cos(\Theta)$$

where

- $r$  (SI unit: m) is the radial distance from Earth's center,
- $L$  (dimensionless) is the maximum degree of the expansion,
- $\Phi$  (SI unit: rad) is the East longitude,
- $\Theta$  (SI unit: rad) is the co-latitude (polar angle),
- $R_e = 6371.2$  km is Earth's radius,
- $g_l^m$  and  $h_l^m$  (SI unit: T) are Gauss coefficients, and
- $P_l^m \cos(\Theta)$  are the Schmidt normalized associated Legendre functions of degree  $l$  and order  $m$ .

### *Collisional Force Theory*

---

It is possible to apply collisional forces that are deterministic or random. A deterministic force based on a friction model can be applied using the [Friction Force](#). Random collisional forces can be applied using the [Collisions](#) node, which includes several different forces for modeling the interaction of particles with a background gas.

#### **FRICION FORCE**

The [Friction Force](#) feature adds the following contribution to  $\mathbf{F}_t$ :

$$\mathbf{F}_t = -m_p \nu (\mathbf{v} - \mathbf{u})$$



The limitations of this option are described in the [Introduction to the Charged Particle Tracing Interface Theory](#) section.

---

where  $\mathbf{v}$  (SI unit: m/s) is the particle velocity and  $\mathbf{u}$  (SI unit: m/s) is the fluid velocity. The collision frequency  $\nu$  (SI unit: 1/s) is either specified directly, or via a collision cross section  $\sigma$  (SI unit: m<sup>2</sup>) and background number density  $N_d$  (SI unit: 1/m<sup>3</sup>):

$$\nu = N_d \sigma |\mathbf{v} - \mathbf{u}|$$

## RANDOM COLLISIONAL FORCES

The **Collisions** node may cause the particle velocity to change during a time step of size  $\Delta t$  taken by the solver, with a collision probability  $P$  defined as:

$$P = 1 - \exp(-\nu\Delta t)$$

If  $P$  is greater than a uniformly distributed random number between 0 and 1 then the particle velocity is reinitialized. The collision frequency  $\nu$  is the sum of the individual collision frequency values  $\nu_j$  for each subnode of the **Collisions** node, which corresponds to a specific type of collision:

$$\nu = \sum_{j=1}^N \nu_j$$

where  $N$  is the total number of subnodes. If two or more subnodes are present and a collision occurs, one of the collision types is chosen at random. The probability  $p_j$  of a specific collision type occurring is

$$p_j = \frac{\nu_j}{\nu}$$

The collision probability for each collision type is

$$\nu_j = N_d \sigma |\mathbf{g}|$$

where  $\mathbf{g} = \mathbf{v} - \mathbf{v}_g$  is the relative velocity in the center of mass reference frame.

The **Collision cross section**  $\sigma$  (SI unit:  $\text{m}^2$ ) is defined for each subnode to the **Collisions** node. The **Background number density**  $N_d$  (SI unit:  $1/\text{m}^3$ ) can either be defined individually for each subnode or inherited from the parent node.

For the purpose of computing the collision probability and post-collision particle velocity, the background gas is assumed to follow a drifting Maxwellian velocity distribution:

$$f(\mathbf{v}_g) = \frac{1}{(2\pi k_B T / m_g)^{3/2}} \exp\left[-\frac{m_g |\mathbf{v}_g - \mathbf{u}|^2}{2k_B T}\right]$$

where:

- $\mathbf{v}_g$  (SI unit:  $\text{m/s}$ ) is the velocity of a gas molecule,
- $\mathbf{u}$  (SI unit:  $\text{m/s}$ ) is the average gas velocity,

- $m_g$  (SI unit: kg) is the mass of a gas molecule,
- $T$  (SI unit: K) is the gas temperature, and
- $k_B = 1.3806488 \times 10^{-23}$  J/K is the Boltzmann constant.

The mass  $m_g$  can be expressed in terms of the molar mass  $M_g$  (SI unit: kg/mol):

$$m_g = \frac{M_g}{N_A}$$

where  $N_A = 6.02214129 \times 10^{23}$  1/mol is the Avogadro constant. The molar mass can either be defined individually for each subnode or inherited from the parent node.

Although all collision types use the same expression for the background gas velocity distribution and the collision probability, they generally have different expressions for the post-collision particle velocity. In addition, some collision types produce secondary particles. The available collision types are described in the following sections.

### ELASTIC

The [Elastic](#) collision force causes a particle to collide with a background gas molecule in such a way that the total energy of the system is conserved. The post-collision velocity of the particle is defined by the expression:

$$\mathbf{v}' = \mathbf{v} - m_g / (m_p + m_g) (\mathbf{g} - \mathbf{g}')$$

where  $\mathbf{g} = \mathbf{v} - \mathbf{v}_g$  is the relative velocity in the center of mass reference frame,  $m_g$  is the mass of the background gas atoms or molecules. The post-collision relative velocity  $\mathbf{g}'$  is  $\mathbf{g}' = |\mathbf{g}|\mathbf{R}$  where  $\mathbf{R}$  is a uniformly distributed random unit vector. No secondary particles are produced.

### EXCITATION

The [Excitation](#) collision force causes a particle to collide with a background gas molecule in such a way that the total energy of the system is not conserved. The post-collision velocity of the particle is defined by the expression:

$$\mathbf{v}' = \mathbf{v} - m_g / (m_p + m_g) (\mathbf{g} - \mathbf{g}')$$

where  $\mathbf{g}' = |\mathbf{g}'|\mathbf{R}$  and where

$$|\mathbf{g}'| = \sqrt{\mathbf{g} \cdot \mathbf{g} - \frac{2\Delta E(m_p + m_g)}{m_p m_g}}$$

where  $\mathbf{R}$  is a uniformly distributed random unit vector, and  $\Delta E$  (SI unit: J) is the kinetic energy lost as a result of the collision. No secondary particles are produced.

#### ATTACHMENT

The [Attachment](#) node causes the model particle to be annihilated, as in the [Disappear](#) condition for the [Wall](#) node. Optionally, the attached species can be released as a secondary particle that is initially at rest.

#### IONIZATION

The [Ionization](#) node causes an electron to collide with a background gas molecule in such a way that the total energy of the system is not conserved. The post-collision velocity of the particle is defined by the expression:

collision velocity of the particle is defined by the expression:

$$\mathbf{v}' = \mathbf{v} - m_g / (m_p + m_g) (\mathbf{g} - \mathbf{g}')$$

where  $\mathbf{g}' = |\mathbf{g}'| \mathbf{R}$  and where

$$|\mathbf{g}'| = \sqrt{\mathbf{g} \cdot \mathbf{g} - \frac{2\Delta E(m_p + m_g)}{m_p m_g}}$$

where  $\mathbf{R}$  is a uniformly distributed random unit vector, and  $\Delta E$  (SI unit: J) is the kinetic energy lost as a result of the collision. No secondary particles are produced.

The background gas molecule is assumed to be ionized; that is, one electron collides with the background gas molecule and causes a secondary electron to be released. For the results to be physical, this particle should inherit its properties from a [Particle Properties](#) or [Override Properties](#) node that is appropriate for electrons.

It is also possible to release the ionized particle as a secondary particle. Both secondary particles, the ion and the electron, are initially assumed to be at rest.

#### CHARGE EXCHANGE COLLISIONS

The [Resonant Charge Exchange](#) and [Nonresonant Charge Exchange](#) nodes are used to model collisions in which charge is exchanged between a model particle and a background gas particle. It is possible to compute the post-collision trajectories of either or both products of the charge exchange reaction. The post-collision velocity of the ionized species  $\mathbf{v}'$  and neutralized species  $\mathbf{v}_n'$  are defined by the expressions:

$$\mathbf{v}' = \mathbf{v}_g + m_g / (m_p + m_g) (\mathbf{g} - \mathbf{g}')$$



$$\mathbf{v}_n' = \mathbf{v} - m_g/(m_p + m_g)(\mathbf{g} - \mathbf{g}')$$

where

$$\mathbf{g}' = \mathbf{g}_\perp \cos(\chi) + \mathbf{g}_\parallel \sin(\chi)$$

and  $\chi$  (SI unit: rad) is the scattering angle in the center of mass coordinate system. The vectors  $\mathbf{g}_\parallel$  and  $\mathbf{g}_\perp$  are defined by the expressions:

$$\mathbf{g}_\parallel = \frac{|\mathbf{g}'|}{|\mathbf{g}|} \mathbf{g}$$

$$\mathbf{g}_\perp = |\mathbf{g}'|(\mathbf{e}_1 \cos(\phi) + \mathbf{e}_2 \sin(\phi))$$

where  $\phi$  (SI unit: rad) is a uniformly distributed angle between 0 and  $2\pi$ . The unit vectors  $\mathbf{e}_1$  and  $\mathbf{e}_2$  are defined so that they form an orthonormal basis with the normalized pre-collision relative velocity  $\mathbf{g}/|\mathbf{g}|$ . The post-collision relative velocity magnitude is defined by the expression:

$$|\mathbf{g}'| = \sqrt{\mathbf{g} \cdot \mathbf{g} - \frac{2\Delta E(m_p + m_g)}{m_p m_g}}$$

where  $\Delta E$  (SI unit: J) is the energy loss. During resonant charge exchange collisions the energy loss is 0 and  $|\mathbf{g}'| = |\mathbf{g}|$ .

### *Particle-Matter Interaction Theory*

---

The [Particle-Matter Interactions](#) node is used to model interaction of energetic ions with solid material. The interaction of energetic ions with the target material is divided into two main interaction types: ionization losses and nuclear stopping.

#### **IONIZATION LOSS**

The [Ionization Loss](#) node models the interaction of ions with the electrons in the target material as a continuous braking force:

$$\mathbf{F} = -S_e \rho \frac{\mathbf{v}}{|\mathbf{v}|}$$

where

- $\mathbf{F}$  (SI unit: N) is the force on the ion,
- $S_e$  (SI unit:  $\text{m}^4/\text{s}^2$ ) is the electronic stopping power,

- $\rho$  (SI unit:  $\text{kg}/\text{m}^3$ ) is the mass density of the target material, and
- $\mathbf{v}$  (SI unit:  $\text{m}/\text{s}$ ) is the ion velocity.

Thus the force always acts opposite the direction of particle motion. For built-in ionization loss models empirical data from [Ref. 1](#) is used to generate a 1D interpolation function, from which the stopping power is expressed as a function of the particle kinetic energy.

### NUCLEAR STOPPING

The [Nuclear Stopping](#) node models the interaction of ions with the nuclei in the target material. Unlike ionization losses, which are treated as a force that is continuous as a function of time, interactions with target nuclei are treated as distinct events that occur instantaneously with a given probability during each time step. In addition, nuclear interactions may change the direction of the ion velocity as well as its magnitude.

During each time step taken by the solver, a value of the scattering angle  $\chi$  (SI unit: rad) is computed for each particle using the expression

$$\chi = \pi - 2 \int_{\xi_{\min}}^{\infty} \frac{b}{\xi^2 \sqrt{1 - \frac{\Phi(\xi)}{\xi \varepsilon} - \frac{b^2}{\xi^2}}} d\xi \quad (4-5)$$

which is a dimensionless version of the expression for the scattering angle as given in [Ref. 3](#), in which  $b$  (dimensionless) is the reduced impact parameter and  $\xi$  is the dimensionless energy defined as

$$\xi = \frac{r}{a_I}$$

where  $r$  (SI unit: m) is the radial distance from the particle trajectory to the target nucleus and  $a_I$  (SI unit: m) is the screening length. The definition of the screening length changes depending on the option selected from the **Screening function** list; see [Table 4-2](#) below.

The reduced energy  $\varepsilon$  (dimensionless) is defined by the expression

$$\varepsilon = \frac{4\pi\varepsilon_0}{Z_p Z_m e^2} a_I E_{cm}$$

Where

- $Z_p$  (dimensionless) is the atomic number of the propagating ions,
- $Z_m$  (dimensionless) is the atomic number of the material,
- $\epsilon_0 = 8.854187817 \times 10^{-12}$  F/m is the permittivity of vacuum,
- $e = 1.602176565 \times 10^{-19}$  C is the elementary charge, and
- $E_{cm}$  (SI unit: J) is the kinetic energy in the center-of-mass coordinate system.

The lower limit of integration  $\xi_{min}$  (dimensionless) is the largest positive root of the equation

$$-\xi_{min}^2 + \frac{\Phi(\xi_{min})}{\epsilon} \xi_{min} + b^2 = 0$$

The screening function  $\Phi(\chi)$  (dimensionless) changes depending on the option selected from the **Screening function** list. The available options are tabulated below.

TABLE 4-2: SCREENING LENGTH AND SCREENING FUNCTION DEFINITIONS

SCREENING FUNCTION	SCREENING LENGTH EXPRESSION $a_I$	SCREENING FUNCTION EXPRESSION $\Phi(\xi)$
None	$\frac{0.8853a_0}{Z_p^{0.23} + Z_m^{0.23}}$	1
Bohr	$\frac{a_0}{Z_p^{0.23} + Z_m^{0.23}}$	$\exp(-\xi)$
Moliere	$\frac{0.8853a_0}{Z_p^{2/3} + Z_m^{2/3}}$	$0.35\exp(-0.3\xi) + 5.5\exp(-1.2\xi) + 0.1\exp(-6\xi)$
Lenz-Jensen	$\frac{0.8853a_0}{Z_p^{2/3} + Z_m^{2/3}}$	$0.7466\exp(-1.038\xi) + 0.2433\exp(-0.3876\xi) + 0.01818\exp(-0.206\xi)$
Universal	$\frac{0.8853a_0}{Z_p^{0.23} + Z_m^{0.23}}$	$0.1818\exp(-3.2\xi) + 0.5099\exp(-0.9423\xi) + 0.2802\exp(-0.4028\xi) + 0.2817\exp(-0.2016\xi)$

Information on the various screening functions and supplemental references can be found in [Ref. 4](#).

The value of the reduced impact parameter  $b$  is sampled from a Rayleigh probability distribution using the expression

$$b = \frac{1}{a_I \lambda} \sqrt{\frac{-\ln(U)}{\pi L N}}$$

where  $U$  is a dimensionless random number sampled with uniform probability from the interval  $(0,1)$ ,  $L$  is the distance traveled by the particle during the time step, and  $N$  is the number density of particles in the target material. Thus it is clear that if very small time steps are taken by the solver, the value of  $b$  is typically quite large. This in turn means that the scattering angle  $\chi$  is very close to 0, so most collisions tend not to have a large effect on the particle trajectory. If the computed value of  $\chi$  is less than the specified **Cutoff scattering angle**  $\chi_c$ , then the collision is deemed insignificant and the particle velocity is not reinitialized during that time step.

If the collision is considered significant, that is if  $\chi \geq \chi_c$ , then in addition to being deflected by the scattering angle, the reinitialized particle trajectory is also rotated by an azimuthal angle  $\phi$  that is sampled at random from the interval  $[0,2\pi]$ . A fraction of the particle energy is also lost to the surroundings; the kinetic energy  $E$  of the particle decreases by the recoil energy  $T$ , defined by the expression

$$T = \frac{4m_1 m_2}{(m_1 + m_2)^2} E \sin^2\left(\frac{\chi}{2}\right)$$

where  $m_1$  and  $m_2$  (SI unit: kg) are the ion mass and the atomic mass of the target material.

To avoid the computational cost of evaluating the integral in [Equation 4-5](#) at every time step for every particle, the value of this integral is tabulated for a range of values of  $b$  and  $\epsilon$ , then imported into models as a set of 2D interpolation functions. Thus, there is a finite range of values in which the nuclear stopping data is computed accurately, corresponding to the interval defined by the inequalities

$$-5 \leq \log(\epsilon) \leq 15$$

$$-25 \leq \log(b) \leq 10$$

outside of this range, the value of the scattering angle is computed by extrapolation and may be less accurate compared to values that are computed within this range.

### *Particle Beam Theory*

---

The [Particle Beam](#) feature can be used to release nonlaminar particle beams with specified emittance values. The classification of a particle beam as laminar or

nonlaminar depends on the distribution of transverse position and velocity within the beam. Further discussion of laminar and nonlaminar beams can be found in [Ref. 5](#).

A beam is laminar if there is a one-to-one relationship between transverse position and velocity. The volume occupied by the beam particles in phase space is essentially zero. In addition, the transverse velocity of particles in a laminar beam must be linearly proportional to the distance from the axis of beam symmetry; otherwise it is possible for particles with unequal transverse velocity components to cross at a later point.

If the beam is nonlaminar, it is possible for the trajectories of particles with different transverse position and velocity components to intersect. Often, such beams occupy nonzero areas in phase space. The area a beam occupies in phase space is related to a quantity known as the emittance that can be defined in several different ways.

The particle beam is released normal to a surface and includes several options for specifying the transverse velocity distribution of beam particles.

#### DEFINITIONS OF BEAM PROPERTIES

If at least one [Particle Beam](#) node is present, additional global variables are defined for certain beam properties. Quantities that indicate the distribution of transverse beam position and velocity are defined in a coordinate system that is centered at the average position of beam particles,  $\mathbf{q}_{\text{av}}$ . In 2D, the transverse direction is orthogonal to the average particle velocity  $\mathbf{v}_{\text{av}}$ ; in 3D, the two transverse directions are orthogonal to  $\mathbf{v}_{\text{av}}$  and to each other.

Let the transverse displacement from the beam center be denoted  $x$ , and the transverse velocity be expressed using the dimensionless variable  $x'$ , which is the ratio of transverse velocity to axial velocity. For 3D, the following discussion can be extended to consider two distinct transverse displacement components in orthogonal directions, called for example  $x_1$  and  $x_2$ .

The area of the phase space ellipse is indicated by the beam emittance. In general, beams may exhibit a gradual fall-off in particle number density, so that it is not always possible to define a sharp boundary where the phase space distribution begins and ends. Instead of being treated as the area of a clearly defined geometric entity, the phase space ellipse can be defined in a statistical sense. The 1-rms emittance  $\epsilon_{1,\text{rms}}$  (SI unit: m) is defined as:

$$\epsilon_{1,\text{rms}} = \sqrt{\langle x^2 \rangle \langle (x')^2 \rangle - \langle xx' \rangle^2}$$

Where the brackets represent an arithmetic mean over all particles. In addition, the 4-rms emittance  $\epsilon_{4,\text{rms}}$  is frequently reported because it corresponds to the area of an ellipse if the distribution of particles in phase space is uniform:

$$\epsilon_{4,\text{rms}} = 4\epsilon_{1,\text{rms}}$$

In addition to the size of the phase space ellipse, several characteristics of its shape can be described using the Twiss parameters  $\alpha$ ,  $\beta$ , and  $\gamma$ , defined as:

$$\alpha = -\frac{\langle xx' \rangle}{\epsilon_{1,\text{rms}}}$$

$$\beta = \frac{\langle x^2 \rangle}{\epsilon_{1,\text{rms}}}$$

$$\gamma = \frac{\langle (x')^2 \rangle}{\epsilon_{1,\text{rms}}}$$

In 3D, it is also possible to define the hyperemittance:

$$\epsilon_h = \epsilon_1 \epsilon_2$$

Depending on the way in which the hyperemittance is defined, either the 1-rms or 4-rms emittance may be used.

### RELEASING PARTICLE BEAMS

The [Particle Beam](#) node includes built-in options for releasing distributions of particles in velocity space by sampling from uniform or Gaussian distributions in each transverse velocity direction. These distributions can be upright if the initial value of the Twiss parameter  $\alpha$  is set to 0. In this case, for the elliptical distributions of particles in phase space, the semimajor and semiminor axes of the ellipse are initially parallel to the  $x$ - and  $x'$ -axes. The initial values of the Twiss parameters must fulfill the Courant-Snyder condition,

$$\gamma\beta - \alpha^2 = 1$$

Thus, out of the three Twiss parameters, it is only necessary to specify  $\beta$  and  $\alpha$ .

Given the initial value of the Twiss parameter  $\beta$ , the 1-rms beam emittance  $\epsilon$  and the Twiss parameter  $\alpha$ , the initial distribution of particles in phase space depends on the option selected from the **Sampling from phase space ellipse** list.

### Sampling from Phase Space

The initial particle positions and velocities can be generated by sampling from the following function, as described in [Ref. 6](#):

$$A^2 = \left(\frac{x}{a}\right)^2 + \left(\frac{ax' - a'x}{\varepsilon_x}\right)^2 + \left(\frac{y}{b}\right)^2 + \left(\frac{by' - b'y}{\varepsilon_y}\right)^2$$

where  $a$  and  $b$  are the semi-major and semi-minor axis of the ellipse in physical space,  $a'$  and  $b'$  are the envelope angles which are related to the Twiss parameters according to:

$$a = 2\sqrt{\varepsilon_x\beta}, \quad b = 2\sqrt{\varepsilon_y\beta}$$
$$a' = -2\alpha\sqrt{\frac{\varepsilon_x}{\beta}}, \quad b' = -2\alpha\sqrt{\frac{\varepsilon_y}{\beta}}.$$

The amplitude of  $A^2$  can be resolved into two components:

$$A^2 = A_x^2 + A_y^2$$

where:

$$A_x^2 = \left(\frac{x}{a}\right)^2 + \left(\frac{ax' - a'x}{\varepsilon_x}\right)^2$$
$$A_y^2 = \left(\frac{y}{b}\right)^2 + \left(\frac{by' - b'y}{\varepsilon_y}\right)^2.$$

#### Algorithm for generating initial positions and velocities

As described in [Ref. 6](#), start by defining two uniformly-distributed random numbers  $\hat{u}_A \in [0, 1]$  and  $\hat{u}_\phi \in [0, 1]$ , then define:

$$A_x = A\sqrt{\hat{u}_\phi}, \quad A_y = A\sqrt{1 - \hat{u}_\phi}$$

with the definition of  $A$  depends on the type of distribution and follows later. Now define two additional uniformly-distributed random numbers  $\hat{\beta}_x \in [0, 1]$  and  $\hat{\beta}_y \in [0, 1]$  then the relative initial positions are the particles are given by:

$$x = aA_x \cos(2\pi\hat{\beta}_x), \quad y = bA_y \cos(2\pi\hat{\beta}_y)$$

and the relative initial transverse velocities by:

$$x' = A_x \left( a' \cos(2\pi \hat{\beta}_x) - \frac{4\epsilon_x}{a} \sin(2\pi \hat{\beta}_x) \right)$$

$$y' = A_y \left( b' \cos(2\pi \hat{\beta}_y) - \frac{4\epsilon_y}{b} \sin(2\pi \hat{\beta}_y) \right)$$

$$z' = \sqrt{1 - x'^2 - y'^2}.$$

The definition of  $A$  changes depending on the requested distribution:

TABLE 4-3: DEFINITION OF VARIABLE A FOR DIFFERENT DISTRIBUTION TYPES

KV	WATERBAG	PARABOLIC
$A = 1$	$A = \sqrt{\frac{3}{2}} \sqrt{\hat{u}_A}$	$A = \sqrt{1 - 2 \cos\left(\frac{\eta - 2\pi}{3}\right)}$ $\eta = \arccos(1 - 2\hat{u}_A)$

The Gaussian distribution is generated in a more straightforward way. Let  $\hat{g}_x, \hat{g}_y, \hat{g}_{x'}, \hat{g}_{y'}$  be normally distributed random numbers with zero mean and standard deviation one. The beam position and velocity can be defined as:

$$x = \frac{a}{2} \hat{g}_x, \quad y = \frac{b}{2} \hat{g}_y$$

and

$$x' = \frac{a'}{a} x + \frac{4\epsilon_x}{2a} \hat{g}_{x'}, \quad y' = \frac{b'}{b} y + \frac{4\epsilon_y}{2b} \hat{g}_{y'}, \quad z' = \sqrt{1 - x'^2 - y'^2}.$$

Once the relative initial positions and velocities are generated, they are converted to global coordinates using the following:

$$\mathbf{r} = \mathbf{r}_c + x\mathbf{t}_1 + y\mathbf{t}_2$$

$$\mathbf{v} = x'V\mathbf{t}_1 + y'V\mathbf{t}_2 + z'V\mathbf{n}$$

where  $\mathbf{r}_c$  is the beam centroid,  $\mathbf{t}_1$  and  $\mathbf{t}_2$  are the two tangent vectors on the surface,  $\mathbf{n}$  is the surface normal and  $V$  is the velocity magnitude.



When the **Beam symmetry** is **Symmetric**, the same initial values of the Twiss parameter  $\beta$ , Twiss parameter  $\alpha$  and beam emittance  $\epsilon$  are applied to each of the two transverse directions. When the **Beam symmetry** is **Asymmetric**, distinct values of these parameters can be assigned to each transverse direction.

When the **Particle release specification** property is **Specify current**, the **Beam symmetry** is **Symmetric**, the **Transverse velocity distribution specification** is **Specify emittance and Twiss parameters** and the **Emittance specification** is **Specify brightness**, the **Brightness** is entered instead of the emittance. In this case, the 1-rms emittance is computed from the **Brightness**  $B$  (SI unit: A/m<sup>2</sup>) and the **Release current magnitude**  $I$  (SI unit: A) using

$$\epsilon_{\text{rms}} = \sqrt{\frac{I}{B}}.$$

*Converting from Phase Space Ellipse Dimensions to Twiss Parameters*

When the **Transverse velocity distribution specification** is set to **Specify phase space ellipse dimensions**, the Twiss parameters are computed from the **Maximum transverse displacement**,  $x_m$ , the **Maximum relative transverse velocity**,  $x'_m$  and the **Rotation angle**,  $\theta$  using the following

$$\beta = \frac{x_m}{x'_m}$$

and

$$\epsilon_{\text{rms}} = \frac{x_m x'_m}{4}.$$

The Twiss parameter  $\alpha$  is computed by first defining a rotation matrix

$$M = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} = \begin{bmatrix} m_{11} & m_{12} \\ m_{21} & m_{22} \end{bmatrix}$$

then defining the semimajor and semiminor axis using

$$x_{m,1} = \frac{x_m}{\sqrt{\frac{x_m^2}{2} + x_m'^2}} \sqrt{\epsilon_{\text{rms}}}$$

$$x'_{m,1} = \frac{x'_m}{\sqrt{\frac{x_m^2}{2} + x_m'^2}} \sqrt{\epsilon_{\text{rms}}}$$

which results in

$$\alpha = -\frac{x'_{m,1}}{x_{m,1}}m_{12}m_{22} - \frac{x_{m,1}}{x'_{m,1}}m_{11}m_{21}.$$

### *Thermionic Emission Theory*

---

The [Thermionic Emission](#) feature is used to release electrons from a hot cathode. It releases model particles from a boundary such that the emitted current density magnitude  $J_{\text{th}}$  (SI unit: A/m<sup>2</sup>) is (Ref. 7)

$$J_{\text{th}} = A^* T^2 \exp\left(-\frac{e\Phi}{k_{\text{B}}T}\right)$$

where

- $T$  (SI unit: K) is the temperature of the cathode,
- $\Phi$  (SI unit: V) is the work function of the metal,
- $A^*$  (SI unit: A/(m<sup>2</sup> K<sup>2</sup>)) is the effective Richardson constant,
- $k_{\text{B}} = 1.3806488 \times 10^{-23}$  J/K is the Boltzmann constant, and
- $e = 1.602176565 \times 10^{-19}$  C is the elementary charge.

The total current  $I_{\text{th}}$  (SI unit: A) released from the surface is

$$I_{\text{th}} = \iint_{\Omega} J_{\text{th}} d\Omega$$

where the surface integral is taken over the selected boundaries.

#### **INITIAL VELOCITY DIRECTIONS**

The initial particle velocity components are

$$v_{t1} = V \sin \theta \cos \phi$$

$$v_{t2} = V \sin \theta \sin \phi$$

$$v_n = V \cos \theta$$

where  $v_n$  is the velocity component normal to the cathode surface and  $v_{t1}$  and  $v_{t2}$  are the two orthogonal velocity components parallel to the surface. The azimuthal angle  $\phi$  is uniformly distributed in the interval  $[0, 2\pi]$ . The polar angle  $\theta$  is

$$\theta = \text{asin} \sqrt{U_1}$$

where  $U_1$  is a uniformly distributed random number in the interval  $[0, 1]$ . The particle speed  $V$  (SI unit: m/s) can be sampled from a probability distribution function in several different ways, which are described in the following section.

### INITIAL SPEED

The probability distribution function of the initial electron speed is more easily expressed in terms of the normalized kinetic energy  $W$  (dimensionless), defined as

$$W = \frac{m_e}{2k_B T} V^2 \quad (4-6)$$

where  $m_e = 9.10938291 \times 10^{-31}$  kg is the electron mass. The thermal electrons are assumed to be nonrelativistic. The probability distribution function of the normalized kinetic energy is

$$f(W) = W \exp(-W) \quad (4-7)$$

The way in which the initial particle kinetic energy is sampled from this probability distribution function is controlled by the **Weighting of macroparticles** list in the physics feature **Initial Velocity** section.

#### *Uniform Current*

For **Uniform Current** the values of the normalized kinetic energy for the particles are sampled from Equation 4-7 using the method of inverse transform sampling, in which the probability distribution function (PDF) is used to compute a cumulative distribution function (CDF) which is then normalized to unity and inverted (inverse normal CDF). When uniformly distributed random numbers in the interval  $[0,1]$  are used as input to the inverse normal CDF, the resulting set of values follows the PDF.

The CDF of Equation 4-7 is

$$c(W) = \int_0^W W' \exp(-W') dW' = 1 - (W + 1) \exp(-W)$$

which is already normalized, i.e.

$$\lim_{W \rightarrow \infty} (1 - (W + 1) \exp(-W)) = 1$$

The inverse normal CDF is

$$W = -\text{LambertW}(-1, (U_3 - 1)\exp(-1)) - 1$$

where  $\text{LambertW}(-1, x)$  is the  $-1$  branch of the Lambert  $W$  function or product log, defined as the real root of the equation

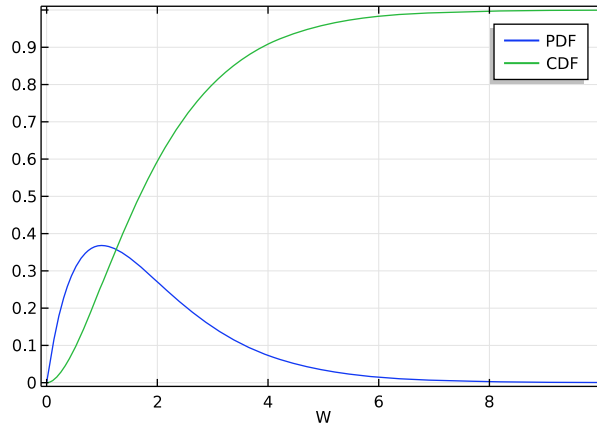
$$\text{LambertW}(-1, x)\exp(\text{LambertW}(-1, x)) = x$$

for which

$$\text{LambertW}(-1, x) \leq -1$$

$U_3$  is a uniform random number in the interval  $[0,1]$  that is uncorrelated with the random numbers used to initialize the polar angle  $\theta$  and azimuthal angle  $\phi$ .

The PDF and CDF are plotted in [Figure 4-1](#).



*Figure 4-1: Normalized probability distribution function (PDF) and cumulative distribution function (CDF) for the normalized particle kinetic energy.*

Because the initial particle energy is sampled from the probability distribution function given in [Equation 4-7](#), each particle carries the same weight; that is, each model particle is a macroparticle representing an equal number of electrons emitted per unit time. This macroparticle weighting is represented by a static degree of freedom stored for all model particles called the effective frequency of release  $f_{\text{rel}}$ ,

$$f_{\text{rel}} = \text{const}$$

To yield the specified total current  $I_{\text{th}}$ , the effective frequency of release of the  $i$ th model particle must be

$$f_{\text{rel},i} = \frac{I_{\text{th}}}{eN}$$

where  $N$  (dimensionless) is the total number of model particles released by the feature.



For more information on the effective frequency of release and its effect on space charge density calculation, see [Space Charge Density Calculation in Theory for the Particle Field Interaction, Non-Relativistic Interface](#).

#### *Uniform Energy Intervals*

When **Uniform energy intervals** is selected, the initial normalized kinetic energy of each particle is uniformly sampled from the interval  $[0, n]$ , where  $n$  is a user-defined proportionality factor. Because the energy is uniformly sampled instead of following the probability distribution given by [Equation 4-7](#), the probability distribution must instead be incorporated into the effective frequency of release:

$$f_{\text{rel}} \propto W \exp(-W) \quad (4-8)$$

Thus the effective frequency of release of the  $i$ th particle is

$$f_{\text{rel},i} = \frac{I_{\text{th}}}{e} \frac{W_i \exp(-W_i)}{\sum W_j \exp(-W_j)}$$

where the sum over  $j$  in the denominator is taken over all model particles released by the feature.

[Equation 4-7](#) has a local maximum at  $W = 1$ , after which it gradually decreases. Some representative values of the CDF are given in [Table 4-4](#).

TABLE 4-4: SELECTED VALUES OF THE CDF FOR NORMALIZED ELECTRON ENERGY

<b>W</b>	<b>CDF</b>
3.89	0.9
6.64	0.99
9.23	0.999

Therefore a value of  $n = 10$ , for example, will encompass more than 99.9% of the cumulative distribution function.

#### *Uniform Speed Intervals*

When **Uniform speed intervals** is selected, the initial speed of each particle is uniformly sampled from the interval  $[0, V_{\text{max}}]$ , where

$$V_{\max} = n \sqrt{\frac{2k_B T}{m_e}}$$

where  $n$  (dimensionless) is a user-defined proportionality factor.

Differentiating both sides of Equation 4-6 and rearranging yields

$$dV = \sqrt{\frac{k_B T}{2m_e W}} dW$$

Substituting into Equation 4-7 yields

$$f(W)dW = f\left(\frac{m_e}{2k_B T} V^2\right) \sqrt{\frac{2m_e W}{k_B T}} dV$$

Therefore, compared to Equation 4-8 the proportionality factor is multiplied by an additional term proportional to  $\sqrt{W}$ :

$$f_{\text{rel}} \propto W^{3/2} \exp(-W) \quad (4-9)$$

Thus the frequency of release of the  $i$ th particle is

$$f_{\text{rel},i} = \frac{I_{\text{th}}}{e} \frac{W_i^{3/2} \exp(-W_i)}{\sum_j W_j^{3/2} \exp(-W_j)}$$

where again the sum over  $j$  in the denominator is taken over all model particle emitted by the feature.

#### *When to Sample from Uniform Energy or Speed Intervals*

The main motivation for selecting **Uniform energy intervals** or **Uniform speed intervals** is that these options devote a disproportionately large number of degrees of freedom to electrons at the extreme ends of the probability distribution function. For example, when sampling from **Uniform energy intervals** with  $n > 8$ , equal representation would be given to particles in the neighborhood of  $W = 1$  and  $W = 7.64$ , despite  $W = 1$  having a probability density 100 times more greater than  $W = 7.64$ .

It can be useful to devote additional degrees of freedom to particles with extremely high and low energy because these particles may have a greater influence on the behavior of a system than electrons of the most probable energy, and therefore they might require finer discretization in velocity space to yield a statistically converged solution. For example, in plane parallel vacuum diodes in the space charge limited

regime, the presence of electrons between the electrodes creates a potential barrier which only the most energetic electrons can bypass. Devoting more degrees of freedom to the most energetic electrons in the distribution, despite the low percentage of the overall number of electrons that they represent, can lead to a more robust and reproducible solution because a larger sample of model particles is used to compute the charge density in the region beyond the potential barrier.

### *Surface Charge Density Theory*

---

The **Surface Charge Density** feature adds an accumulated variable to the boundary elements in the selection of its parent node. The variable defined on one of these boundary elements is incremented by the charge of each particle that hits it, divided by the surface area of the element,

$$\sigma_i = \frac{1}{A_i} \sum eZ$$

where

- $\sigma_i$  (SI unit: C/m<sup>2</sup>) is the surface charge density in boundary element  $i$ ,
- $A_i$  (SI unit: m<sup>2</sup>) is the area of the boundary element,
- $e = 1.602176565 \times 10^{-19}$  C is the elementary charge, and
- $Z$  (dimensionless) is the charge number.

The sum is taken over all particles that hit the boundary element.

### *Current Density Theory*

---

The **Current Density** feature adds one or more accumulated variables to each boundary element in the selection of its parent node.

If the **Type** is set to **Normal current density**, the normal current density in each boundary element is

$$J_{n,i} = \frac{1}{A_i} \sum f_{\text{rel}} eZ \cos \phi$$

where

- $J_{n,i}$  (SI unit: A/m<sup>2</sup>) is the normal current density in boundary element  $i$ ,
- $A_i$  (SI unit: m<sup>2</sup>) is the area of the boundary element,

- $e = 1.602176565 \times 10^{-19}$  C is the elementary charge,
- $Z$  (dimensionless) is the charge number,
- $\phi$  (SI unit: rad) is the acute angle of incidence measured from the surface normal, and
- $f_{\text{rel}}$  (SI unit: 1/s) is the effective frequency of release.

The sum is taken over all particles that hit the boundary element. A particle's effective frequency of release is determined by the current of the particle's release feature and the total number of particles released by the feature,

$$(f_{\text{rel}})_i = \frac{I}{N_{\text{tf}}eZ}$$

If the **Type** is set to **Current density**, the components of the current density vector  $\mathbf{J}_{s,i}$  are then calculated in each boundary element:

$$\mathbf{J}_{s,i} = \frac{1}{A_i} \sum f_{\text{rel}} e Z \frac{\mathbf{v}}{V}$$

where  $\mathbf{v}/V$  is a unit vector in the same direction as the velocity of the incident particle.

### *Heat Source Theory*

---

The **Heat Source** feature adds an accumulated variable to the boundary elements in the selection of its parent node. This accumulated variable represents a boundary heat source equal to the kinetic energy lost by incident particles per unit time, divided by the surface area of the boundary element:

$$Q_i = \frac{1}{A_i} \sum f_{\text{rel}} E$$

where

- $Q_i$  (SI unit: W/m<sup>2</sup>) is the heat source per unit area in boundary element  $i$ ,
- $A_i$  (SI unit: m<sup>2</sup>) is the surface area of the element,
- $f_{\text{rel}}$  (SI unit: 1/s) is the effective frequency of release of the incident particle, and
- $E$  (SI unit: J) is the particle kinetic energy.

The sum is taken over all particles that hit the boundary element.



## Etch Theory

---

The **Etch** feature uses accumulated variables to calculate the etch rate due to physical sputtering by energetic particles on a surface. The dependence of the sputtering yield on the energy of particles and their angle of incidence has been described by Yin and Sawin (Ref. 8) and by Guo and Sawin (Ref. 9).

If the **Plasma type** is **Collisionless**, a single accumulated variable is defined on each boundary element in the selection of the parent node. The etch rate (SI unit: m/s) is defined as

$$E_{E,i} = \frac{M_s}{\rho N_A A_i} \sum f_{\text{rel}} Y(\phi) \left( \sqrt{\frac{E}{Y_s}} - \sqrt{\frac{E_{\text{th}}}{Y_s}} \right)$$

where

- $M_s$  (SI unit: kg/mol) is the molar mass of the surface material,
- $\rho$  (SI unit: kg/m<sup>3</sup>) is the density of the surface material,
- $N_A = 6.02214129 \times 10^{23}$  1/mol is Avogadro's constant,
- $A_i$  (SI unit: m<sup>2</sup>) is the surface area of the boundary element,
- $f_{\text{rel}}$  (SI unit: 1/s) is the effective frequency of release,
- $Y(\phi)$  (dimensionless) is a function that indicates the dependence of the etch rate on the angle of incidence  $\phi$ ,
- $E$  (SI unit: J) is the particle kinetic energy,
- $E_{\text{th}}$  (SI unit: J) is the threshold energy, and
- $Y_s$  (SI unit: J) is the slope of the etch yield curve.

The sum is taken over all particles that hit the boundary element.

If the **Plasma type** is **Collisional**, the effective frequency of release is no longer a reliable indicator of the incident charged particle current. Instead, incident particles in each boundary element are assigned a weighting factor  $\Gamma_i$  (SI unit: 1/(m<sup>2</sup>s)), defined as

$$\Gamma_i = \frac{\mathbf{n} \cdot \mathbf{J}_i}{\sum eZ \sin \phi}$$

where the numerator is the normal current of charged particles at the surface. The etch rate is then defined as

$$E_{E,i} = \frac{\Gamma_i M_s}{\rho N_A} \sum Y(\phi) \left( \sqrt{\frac{E}{Y}} - \sqrt{\frac{E_{th}}{Y}} \right)$$

A typical angular dependence function is given by Guo and Sawin (Ref. 9) as

$$Y(\phi) = -141.3 \cos^6 \phi + 641.1 \cos^5 \phi - 1111.3 \cos^4 \phi + 944.6 \cos^3 \phi - 422.0 \cos^2 \phi + 95.31 \cos \phi - 5.460$$


where the angle of incidence  $\phi$  is measured from the surface normal. This angular dependence function is appropriate for physical sputtering of polysilicon by argon ions.

### *References for the Charged Particle Tracing Interface*

---

1. NIST Stopping-Power and Range Tables for Electrons, Protons, and Helium Ions, <http://www.nist.gov/pml/data/star/index.cfm>.
2. International Geomagnetic Reference Field (IGRF), <http://www.ngdc.noaa.gov/AGA/vmod/igrf.html>.
3. L.D. Landau and E.M. Lifshitz, *Mechanics*, 3rd ed., Elsevier, 1976.
4. J. F. Ziegler, J. P. Biersack, and M. D. Ziegler, *The Stopping and Range of Ions and Matter*, SRIM, 2015.
5. S. Humphries, *Charged Particle Beams*, Dover, 2013.
6. S. M. Lund, T. Kikuchi, R. C. Davidson, “Generation of initial Vlasov distributions for simulation of charged particle beams with high space-charge intensity,” *Physical Review Special Topics--Accelerators and Beams*, 2007.
7. P. T. Kirstein, G. S. Kino, and W. E. Waters, *Space charge flow*, McGraw-Hill, 1967.
8. Y. Yin and H.H. Sawin, “Surface Roughening of Silicon, Thermal Silicon Dioxide, and Low-k Dielectric Coral Films in Argon Plasma,” *J. Vac. Sci. Technol. A*, vol. 26, no. 1, pp. 151–160, 2008.
9. W. Guo and H.H. Sawin, “Modeling of the Angular Dependence of Plasma Etching,” *J. Vac. Sci. Technol. A*, vol. 27, no. 6, pp. 1326–1336, 2009.

# Particle Tracing for Fluid Flow



This chapter describes the Particle Tracing for Fluid Flow interface found under the **Fluid Flow** branch () when adding a physics interface.

In this chapter:

- [The Particle Tracing for Fluid Flow Interface](#)
- [Theory for the Particle Tracing for Fluid Flow Interface](#)

See [The Single-Phase Flow, Laminar Flow Interface](#) in the *COMSOL Multiphysics Reference Manual* for other Fluid Flow interface and feature node settings.

# The Particle Tracing for Fluid Flow Interface

The **Particle Tracing for Fluid Flow (ftp)** interface (  ), found under the **Fluid Flow>Particle Tracing** branch (  ) when adding a physics interface, is used to compute the motion of particles in a background fluid. Particle motion can be driven by drag, gravity, electric, magnetic, and acoustophoretic forces. User-defined forces can be added. It is also possible to compute the particle mass and temperature as well as particle-fluid interactions.

When this physics interface is added, these default nodes are also added to the **Model Builder—Wall** and **Particle Properties**. Then, from the **Physics** toolbar, add other nodes that implement, for example, boundary conditions and volume forces. You can also right-click **Particle Tracing for Fluid Flow** to select physics features from the context menu.

## SETTINGS

The **Label** is the default physics interface name.

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 `ftp`.

## PARTICLE RELEASE AND PROPAGATION

The **Formulation**, **Relativistic Correction**, **Store extra time steps for wall interactions**, and **Maximum number of secondary particles** settings are the same as for [The Mathematical Particle Tracing Interface](#), except that only the **Newtonian**, **Newtonian, first order**, and **Massless** formulations are available.

### *Particle Release Specification*

Select an option from the **Particle release specification** list—**Specify release times** (the default) or **Specify mass flow rate**. If **Specify release times** is selected, then each model particle is treated as the instantaneous position of one or more particles for the purpose of modeling fluid-particle interactions. This means, for example, that if the [Volume](#)

[Force Calculation](#) node is used, the volume force on the fluid is only nonzero in mesh elements that are currently occupied by particles.

If **Specify mass flow rate** is selected, then for the purpose of modeling fluid-particle interactions, each model particle traces a path that is followed by a number of particles per unit time. This means that the volume force computed by the [Volume Force Calculation](#) node is nonzero in all mesh elements that the particle trajectories pass through, not just at the instantaneous positions of the particles. In other words, the model particles leave behind a trail of nonzero force components in the mesh elements they pass through.

The **Specify mass flow rate** option is primarily used to model streams of particles under steady-state conditions. Changing the particle release specification affects some inputs in the settings windows for release features such as the [Release](#) and [Inlet](#) nodes. In addition, the [Mass Deposition](#) subnode to the [Wall](#) node is only available with the **Specify release times** option, while the [Boundary Load](#) and [Mass Flux](#) subnodes are only available when **Specify mass flow rate** is selected.

#### ADDITIONAL VARIABLES

The options **Store particle status data** and **Store particle release statistics** are the same as for [The Mathematical Particle Tracing Interface](#). The option **Include out-of-plane degrees of freedom** is the same as for [The Charged Particle Tracing Interface](#).

#### *Compute Particle Temperature*

Select the **Compute particle temperature** check box to compute particle temperatures (the default is to not compute particle temperatures). When this option is activated the temperature of the particle is computed by solving an additional ordinary differential equation per particle. Thermal properties for the particles can be specified in the Settings window for the [Particle Properties](#) node.



By selecting this check box it is possible to add [Heat Source](#), [Convective Heat Losses](#) and [Radiative Heat Losses](#) nodes, which are available from the context menu (right-click the parent node, then view the **Thermal** list) or from the **Physics** toolbar, **Domains** menu.



[Computing Particle Temperature](#) in the theory section.

---

### Compute Particle Mass

Select the **Compute particle mass** check box to compute particle masses (the default is to not compute particle masses). When this option is activated an additional equation for the particle mass is computed by solving an additional ordinary differential equation per particle. The accretion rate of each particle can be specified in the Settings window for the [Particle Properties](#) node.



The **Compute particle mass** check box must be selected in order to use the [Droplet Breakup](#) and [Nozzle](#) features, which are available from the context menu (right-click the parent node). The **Droplet Breakup** feature is also accessible from the **Physics** toolbar, **Domains** menu. The **Nozzle** feature is accessible from the **Physics** toolbar, **Points** and **Global** menus.



[Computing Particle Mass](#) in the theory section.

### Enable Macroparticles

Some fluid-particle systems contain such a large number of particles or droplets that modeling each entity individually is not feasible. In such cases, it is often convenient to introduce the concept of a *macroparticle*, or a single model particle that can represent a larger number of real particles.

Select the **Enable macroparticles** check box to allocate an auxiliary dependent variable for a dimensionless multiplication factor, allowing the number of real particles represented by each model particle to be stored.



The **Enable macroparticles** check box must be selected in order to use the [Nozzle](#) feature, which is available from the context menu (right-click the parent node) or from the **Physics** toolbar, **Points** and **Global** menus.

The **Enable macroparticles** check box is only available if **Specify release times** is selected from the **Particle release specification** list. Otherwise, the weighting of each macroparticle is determined by the **Mass flow rate**, which is specified in the release features.

### Include Rarefaction Effects

Select the **Include rarefaction effects** check box to apply correction factors to the forces defined by the **Drag Force** and **Thermophoretic Force** nodes. These correction factors

improve the accuracy of the drag and thermophoretic forces when the particle Knudsen number is significantly large. This can be used to model the motion of particles in a rarefied gas flow.



Several options, such as **Include out-of-plane degrees of freedom**, **Compute particle mass**, and **Compute particle temperature**, cause auxiliary dependent variables to be allocated for the particles. These auxiliary dependent variables are solved for by defining first-order differential equations. If, in addition, **Newtonian** is selected from the **Formulation** list, then the resulting system of equations includes both first- and second-order equations.

Mixed first- and second-order equations systems are not supported for all solver configurations. For example, it is not possible to use explicit time stepping methods such as Dormand-Prince 5. To use explicit time-stepping methods with auxiliary dependent variables, consider using the **Newtonian, first order** formulation instead.

## ADVANCED SETTINGS

These settings are the same as for [The Mathematical Particle Tracing Interface](#).

## DEPENDENT VARIABLES

The dependent variables (field variables) are the **Particle position**, **Particle position components**, **Particle velocity**, **Particle velocity components**, **Particle temperature**, **Particle mass**, and **Number multiplication factor**. Note that not every dependent variable is needed for every combination of physics interface settings; for example, the **Particle velocity** and **Particle velocity components** are only used if **Newtonian, first order** is selected from the **Formulation** list. The names can be changed but the names of fields and dependent variables must be unique within a model.



- [Domain, Boundary, Pair, and Global Nodes for the Particle Tracing for Fluid Flow Interface](#)
- [Theory for the Mathematical Particle Tracing Interface](#)
- [Newtonian Formulation and Massless Formulation](#)
- [Theory for the Particle Tracing for Fluid Flow Interface](#)



- *Particle Trajectories in a Laminar Static Mixer*: Application Library path **Particle\_Tracing\_Module/Fluid\_Flow/laminar\_mixer\_particle**
- With the CFD Module see *Particle Tracing in a Micromixer*: Application Library path **CFD\_Module/Particle\_Tracing/micromixer\_particle\_tracing**.
- With the Acoustics Module see *Acoustic Levitator*: Application Library path **Acoustics\_Module/Nonlinear\_Acoustics/acoustic\_levitator**.

---

### *Domain, Boundary, Pair, and Global Nodes for the Particle Tracing for Fluid Flow Interface*

---

The [Particle Tracing for Fluid Flow Interface](#) has these domain, boundary, pair, and global nodes available from the **Physics** ribbon toolbar (Windows users), **Physics** context menu (Mac or Linux users), or right-click to access the context menu (all users). The context menu includes dedicated menus for **Forces** and **Thermal** features.



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.



In the *COMSOL Multiphysics Reference Manual* see [Table 2-3](#) for links to common sections and [Table 2-4](#) 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.



For a theoretical background to the forces, see [Theory for the Particle Tracing for Fluid Flow Interface](#).



These nodes and subnodes are described in this section (listed in alphabetical order):

- Acoustophoretic Force
- Boundary Load
- Brownian Force
- Convective Heat Losses
- Dielectrophoretic Force
- Drag Force
- Droplet Breakup
- Electric Force
- Erosion
- Gravity Force
- Heat Source
- Lift Force
- Magnetic Force
- Magnetophoretic Force
- Mass Deposition
- Mass Flux
- Nozzle
- Particle Properties
- Radiative Heat Losses
- Shell
- Symmetry
- Thermophoretic Force
- Volume Force Calculation

These nodes and subnodes are described for [The Mathematical Particle Tracing Interface](#) (listed in alphabetical order):

- Accumulator (Boundary)
- Accumulator (Domain)
- Auxiliary Dependent Variable
- Axial Symmetry
- Force
- Inlet
- Nonlocal Accumulator
- Outlet
- Override Properties
- Particle Continuity
- Particle Counter
- Particle-Particle Interaction
- Periodic Condition
- Release
- Release from Data File
- Release from Edge
- Release from Grid
- Release from Point
- Secondary Emission
- Velocity Reinitialization
- Wall (the default boundary condition)

## Particle Properties

---

Use the **Particle Properties** node to specify the particle properties, charge number, and particle velocity based on whether a Newtonian formulation or massless formulation is selected.

### PARTICLE PROPERTIES

This section is shown when **Newtonian** or **Newtonian, first order** is selected as the **Formulation** for the physics interface. If the **Compute particle mass** check box is cleared in the physics interface **Additional Variables** section, select the **Particle property specification: Specify particle density and diameter** (the default), **Specify particle mass and density**, or **Specify particle mass and diameter**. Then enter the applicable values or expressions for the following:

- **Particle density**  $\rho_p$  (SI unit:  $\text{kg}/\text{m}^3$ ). The default is  $2200 \text{ kg}/\text{m}^3$ .
- **Particle diameter**  $d_p$  (SI unit: m). The default is  $1\text{E}-6$  m.
- **Particle mass**  $m_p$  (SI unit kg). The default is  $1\text{E}-6$  kg.

If the **Compute particle mass** check box is selected, enter a value or expression for the **Particle density**.

Select an option from the **Particle type** list—**Solid particles** (the default) or **Liquid droplets/bubbles**. If **Liquid droplets/bubbles** is selected, enter values or expressions for the following:

- **Particle dynamic viscosity**  $\mu_p$  (SI unit:  $\text{Pa}\cdot\text{s}$ ) the default is  $1 [\text{mPa}\cdot\text{s}]$ .
- **Particle surface tension**  $\sigma_p$  (SI unit:  $\text{N}/\text{m}$ ). The default is  $0.0729 [\text{N}/\text{m}]$ .



**Liquid droplets/bubbles** must be selected for at least one species in the model in order to use the [Droplet Breakup](#) and [Nozzle](#) features.

---

### CHARGE NUMBER

Enter a value or expression for the **Charge number**  $Z$  (dimensionless). The default is 0.

### PARTICLE VELOCITY

This section is shown when **Massless** is selected as the **Formulation** for the physics interface. Enter a vector for the **Particle velocity**  $\mathbf{v}$  (SI unit:  $\text{m}/\text{s}$ ) based on space dimension. The **Massless** formulation means that the particles follow streamlines of the particle velocity expression.

## THERMAL PROPERTIES

This section is shown when the **Compute particle temperature** check box is selected in the physics interface **Additional Variables** section. Enter a value or expression for the **Particle specific heat capacity**  $C_p$  (SI unit: J/(kg K)). The default value is 2200[J/(kg\*K)]. Then enter a value or expression for the **Particle latent heat**  $h_p$  (SI unit: J/kg). The default value is 42000[J/kg].

## ACCRETION RATE

This section is shown when the **Compute particle mass** check box is selected in the physics interface **Additional Variables** section. Enter a value for the **Accretion rate**  $R$  (SI unit: kg/s). The default is 0. The **Accretion rate** is the time derivative of the particle mass.

## *Symmetry*

---

Use the **Symmetry** node to define a plane of symmetry at selected boundaries in the geometry.

For particle tracing models, the physical interpretation of plane symmetry is as follows: for every particle that exits the domain through a boundary assigned a **Symmetry** node, a new particle enters the modeling domain at the same location at the same time, with an incoming velocity that mirrors the outgoing velocity of the exiting particle. Thus the model particle is specularly reflected as if it hit a **Wall** with the **Bounce** condition.



The main advantage of using the **Symmetry** condition instead of a **Wall** feature with the **Bounce** condition is that the **Symmetry** boundaries are ignored for the purposes of computing the distance and direction of each particle from the nearest wall, which is used for wall induced lift, drag, and turbulent dispersion by the **Drag Force** and **Lift Force** features.

---

## *Drag Force*

---



The **Drag Force** node is only available when **Newtonian** or **Newtonian, first order** is selected as the **Formulation** on the **Settings** window for **The Particle Tracing for Fluid Flow Interface**.

---

Use the **Drag Force** node to select a drag law, specify the velocity field, enter material properties, and optionally apply random perturbations to account for particle motion in a turbulent flow.

### DRAG FORCE

Select a **Drag law**: **Stokes** (the default), **Schiller-Naumann**, **Haider-Levenspiel**, **Oseen correction**, **Hadamard-Rybczinski**, or **Standard drag correlations**.



**Stokes** drag and **Oseen correction** are applicable for particles that have a relative Reynolds number much less than one. Either should be a good enough approximation in most cases. If the particle Reynolds number is greater than one, then select **Schiller-Naumann**. If, in addition, the particles are non-spherical, select **Haider-Levenspiel**. If the particles are very pure gas bubbles or liquid droplets, select **Hadamard-Rybczinski**. The **Standard drag correlations** are a set of piecewise-continuous functions that are applicable over a wide range of relative Reynolds numbers.

- For all choices, enter coordinates for the **Velocity field  $\mathbf{u}$**  (SI unit: m/s) based on space dimension. If another physics interface is present which computes the velocity field then this can be selected from the list.
- For all choices, the **Dynamic viscosity  $\mu$**  (SI unit: Pa·s) is taken **From material**. For **User defined** enter another value or expression.
- If **Stokes** is selected as the **Drag law**, select a **Wall Correction** to include a correction factor to the drag force for particles near walls.
- If **Schiller-Naumann**, **Haider-Levenspiel**, **Oseen correction**, **Hadamard-Rybczinski**, or **Standard drag correlations** is selected as the **Drag law**, or if the **Include rarefaction effects** check box is selected in the physics interface **Particle Release and Propagation** section, the fluid **Density  $\rho$**  (SI unit: kg/m<sup>3</sup>) is taken **From material**. For **User defined** enter another value or expression.
- For **Haider-Levenspiel** enter a value or expression for the particle **Sphericity  $S_p$**  (dimensionless). The default is 1.
- For **Hadamard-Rybczinski** enter a value or expression for the **Particle dynamic viscosity  $\mu_p$**  (SI unit: Pa·s). The default value is  $1.8e-5$  [Pa\*s].



Particle Motion in a Fluid in Theory for the Particle Tracing for Fluid Flow Interface.

## RAREFACTION EFFECTS

This section is only available if the **Include rarefaction effects** check box is selected in the physics interface **Particle Release and Propagation** section. It defines a correction factor that is multiplied by the drag force to account for a large particle Knudsen numbers.

Select an option from the **Rarefaction effects** list: **Basset**, **Epstein**, **Phillips**, **Cunningham-Millikan-Davies** (the default), or **User defined**.



The **Basset** model is appropriate for relative Knudsen numbers much less than one, while the **Epstein** model is appropriate for free molecular flows. The **Phillips** model is usable at intermediate values of the relative Knudsen number and shares the same asymptotic behavior as the **Epstein** and **Basset** numbers at very large and small Knudsen numbers, respectively. The **Cunningham-Millikan-Davies** model includes three tunable parameters that can be used to fit the drag force correction factor to empirical data.

Select an option from the **Mean free path calculation** list—**Ideal gas, hard sphere collisions** (the default) or **User defined correction**. For **User defined correction** enter a **Mean free path correction**  $\lambda'/\lambda$  (dimensionless). The default value is 1.

- If **Basset**, **Epstein**, or **Phillips** is selected from the **Rarefaction effects** list, enter an **Accommodation coefficient**  $\sigma_R$  (dimensionless). The default value is 1. The accommodation coefficient may be interpreted as the fraction of gas molecules that undergo diffuse reflection at the particle surface.
- If **Cunningham-Millikan-Davies** is selected from the **Rarefaction effects** list, enter the three dimensionless coefficients  $C_1$ ,  $C_2$ , and  $C_3$ . The default values are 2.514, 0.8, and 0.55, respectively.

Enter a value or expression for the **Pressure**  $p$  (SI unit: Pa). If a physics interface is present that computes the pressure, it can be selected directly from the list.



[Drag Force Correction Factors in Theory for the Particle Tracing for Fluid Flow Interface.](#)

## INCLUDE WALL CORRECTIONS

This section is only available if **Stokes** is selected as the **Drag law** in the **Drag Force** section. Select the **Include wall corrections** box to define a correction factor that is multiplied by the drag force as particles approach a wall.



Wall Corrections in Theory for the Particle Tracing for Fluid Flow Interface.

## TURBULENT DISPERSION

If particles are moving in a turbulent flow, this section can be used to apply random perturbations to the drag force to account for the turbulence.

Select an option from the **Turbulent dispersion model** list—**None** (the default), **Discrete random walk**, **Discrete random walk, variable time step**, or **Continuous random walk**.

- If **None** is selected, no turbulent dispersion term is applied.
- If **Discrete random walk** is selected, a random term is added to the background fluid velocity when computing the drag force at every time step taken by the solver. The random perturbation is regenerated at every time step.
- If **Discrete random walk, variable time step** is selected, a random term is added to the background fluid velocity when computing the drag force at every time step taken by the solver. Unlike **Discrete random walk**, the random perturbation term is only regenerated at time intervals greater than or equal to the eddy lifetime of the flow. Usually, this improves the statistical convergence of the particle distribution as the time step size is made arbitrarily small.
- If **Continuous random walk** is selected, a random perturbation is applied to each particle by integrating a Langevin equation. Unlike **Discrete random walk**, the perturbation of the background velocity that is applied to each particle depends on the time history of all previously applied perturbations.

If **Discrete random walk**, **Discrete random walk, variable time step**, or **Continuous random walk** is selected, enter a value or expression for the **Turbulent kinetic energy**  $k$  (SI unit:  $\text{m}^2/\text{s}^2$ ), which determines the magnitude of the turbulent dispersion term. If a physics interface is present that computes the turbulent kinetic energy, it can be selected directly from the list.

If **Discrete random walk, variable time step** or **Continuous random walk** is selected, enter a value or expression for the **Turbulent dissipation rate**  $\varepsilon$  (SI unit:  $\text{m}^2/\text{s}^3$ ), which is related to the lifetime of eddy currents in the flow and thus determines how frequently

the random perturbation term is sampled. If a physics interface is present that computes the turbulent dissipation rate, it can be selected directly from the list. Then enter a value or expression for the **Lagrangian time scale coefficient**  $C_L$  (dimensionless). The default value is 0.2.

If **Continuous random walk** is selected, it is possible to adjust the random contributions to the fluid velocity to account for anisotropic turbulence, which is often significant near boundaries. Select the **Include anisotropic turbulence in boundary layers** check box to compute different random contributions to the fluid velocity field based on the streamwise, spanwise, and wall normal directions in the flow, which are computed using the fluid velocity direction and the direction to the nearest point on an adjacent wall. When searching for the nearest wall, boundaries using the [Symmetry](#), [Inlet](#), and [Outlet](#) features are ignored.



[Particle Motion in a Turbulent Flow](#) in [Theory for the Particle Tracing for Fluid Flow Interface](#).

---

#### ADVANCED SETTINGS

Use the **Particles to affect** list to apply the force to specific particles. The available settings are the same as for the [Force](#) node.

If any option except **None** is selected from the **Turbulent dispersion model** list, the **Drag Force** feature creates random numbers. If, in addition, the **Arguments for random number generation** setting is **User defined** in the physics interface **Advanced Settings** section, enter the **Additional input argument to random number generator**. The default value is 1.

#### *Lift Force*

---



The **Lift Force** node is only available when **Newtonian** or **Newtonian, first order** is selected as the **Formulation** on the **Settings** window for [The Particle Tracing for Fluid Flow Interface](#).

---

Use the **Lift Force** node to exert a lift force on particles in a fluid. Lift forces are applicable when particles move through a fluid in which the velocity field is nonuniform.

## LIFT FORCE

Select a **Lift law**: **Saffman** (the default) or **Wall induced**.



**Saffman** and **Wall induced** lift forces are applicable for particles traveling through a creeping flow. The **Saffman** lift force is applicable for particles far from walls. The **Wall induced** lift force is a specialized formulation to account for the effects of nearby walls as particles move through pipes or channels.

Enter coordinates for the **Velocity field  $\mathbf{u}$**  (SI unit: m/s) based on space dimension. If another physics interface is present which computes the velocity field then this can be selected from the list.

The **Dynamic viscosity  $\mu$**  (SI unit: Pa·s) is taken **From material**. For **User defined** enter another value or expression.

If **Saffman** is selected from the **Lift law** list it is only necessary to specify the set of domains in which the lift force is exerted. If **Wall induced** is selected, the **Parallel Boundary 1** and **Parallel Boundary 2** selections are also shown. These two boundary selections are typically two parallel surfaces on either side of a parabolic flow profile.



[Particle Motion in a Shear Flow in Theory for the Particle Tracing for Fluid Flow Interface.](#)

## *Brownian Force*



The **Brownian Force** node is only available when **Newtonian** or **Newtonian, first order** is selected as the **Formulation** on the **Settings** window for [The Particle Tracing for Fluid Flow Interface](#).

Use the **Brownian Force** node to account for diffusion of suspended particles in a fluid according to Einstein's theory. When using this feature, a random force is applied on each particle at each time step. This often forces the solver to take very small time steps



because the default relative and absolute tolerances solver settings are quite strict ( $10^{-5}$  and  $10^{-6}$ , respectively).



In order to make the model solve efficiently, it might be necessary to relax these tolerances. The solver needs to take enough time steps so that the random force is applied enough times to be meaningful, but not so many time steps that the model takes a very long time to solve. As a general rule of thumb, it is usually best to relax both the tolerances to  $10^{-3}$  and then perform a convergence study on the tolerances.

### BROWNIAN FORCE

Enter a value or expression for the fluid **Temperature**  $T$  (SI unit: K). The default is 293.15 K. If another physics interface that computes the fluid temperature is present in the model, then this can be selected from the list as a model input.

Enter a value or expression for the **Dynamic viscosity**  $\mu$  (SI unit: Pa·s). If a physics interface is present that computes the dynamic viscosity it can be selected directly from the list, or the value or expression can come from the selected material.

### ADVANCED SETTINGS

Use the **Particles to affect** list to apply the force to specific particles. The available settings are the same as for the [Force](#) node.

The random number generator used in COMSOL Multiphysics uses seeding so that the same results are obtained every time the model is solved. When adding forces that are random in nature, this is not always desirable. When the **Arguments for random number generation** setting is **User defined**, you can use the **Additional input argument to random number generator** field in conjunction with a Parametric Sweep to perform a Monte Carlo simulation. For example, use the following steps:

- 1 From the **Home** toolbar click **Parameters** (  $P_i$  ). Or in the **Model Builder**, right-click **Global Definitions** (  $\equiv$  ) and select **Parameters**.
- 2 Add a parameter with **Name**, for example,  $ds$  and set the **Expression** to 1.
- 3 In the **Brownian Force** feature, set the **Additional input argument to random number generator** to  $ds$ .
- 4 Add a **Parametric Sweep** to the current **Study** and set the **Parameter name** to  $ds$ .

- 5 Set the **Parameter values** to be an array whose length is equal to the number of times the model should be solved. For example, to solve the model 5 times, set the **Parameter values** to 1 2 3 4 5.



When **2** (the default) is selected from the **Wall accuracy order** list in the physics interface **Advanced Settings** section, the Brownian force may cause particles to undergo extremely large accelerations after they are released or when they interact with walls. This is due to the second-order Runge-Kutta extrapolation using the random force to predict the particle position at the end of the current time step. To prevent this non-physical behavior, consider selecting **1** from the **Wall accuracy order** list when modeling Brownian motion.



- [About Parameters, Variables, and Expressions](#) and [Parametric Sweep](#) in the *COMSOL Multiphysics Reference Manual*
- [Additional Forces](#) in the theory section.



*Brownian Motion*: Application Library path  
**Particle\_Tracing\_Module/Tutorials/brownian\_motion**

## *Gravity Force*



The **Gravity Force** node is only available when **Newtonian** or **Newtonian, first order** is selected as the **Formulation** in the physics interface **Particle Release and Propagation** section.

Use the **Gravity Force** node to define the gravity vector and density. The gravity force causes particles with a suitably high mass and density to move in the same direction as the orientation of the gravity vector. For submicron particles, the drag and other external forces can dominate and gravity can have little effect on the particle trajectories.

### **GRAVITY FORCE**

Enter coordinates based on space dimension for the **Gravity vector  $\mathbf{g}$**  (SI unit:  $\text{m}/\text{s}^2$ ). The default magnitude for the gravity vector is  $\mathbf{g\_const}$ , 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 for the gravity vector is in the  $y$ -direction for 2D components.

---



The default direction for the gravity vector is in the  $z$ -direction for 3D and 2D axisymmetric components.

---

The default **Density**  $\rho$  (SI unit:  $\text{kg/m}^3$ ) is taken **From material**, from a physics interface that defines or computes the density, or select **User defined** to enter another value or expression.

#### AFFECTED PARTICLES

Use the **Particles to affect** list to apply the force to specific particles. The available settings are the same as for the [Force](#) node.



[Particle Motion in a Fluid](#) in the theory section.

---

### *Acoustophoretic Force*

---



The **Acoustophoretic Force** node is only available when **Newtonian** or **Newtonian, first order** is selected as the **Formulation** on the **Settings** window for [The Particle Tracing for Fluid Flow Interface](#).

---

Use the **Acoustophoretic Force** node to add forces on small particles due to acoustic radiation. This force only adds equation contributions if the acoustic pressure was computed from a **Frequency Domain** study type. This is because the magnitude of the acoustophoretic force depends on the angular frequency.

## ACOUSTIC PROPERTIES

Enter a value or expression for the following:

- **Pressure**  $p$  (SI unit: Pa). This should be computed from another physics interface using a **Frequency Domain** study type.
- **Speed of sound**  $c$  (SI unit: m/s) of the surrounding fluid. This can either be taken **From material** or **User defined**, in which case the default is 343 m/s.
- **Density**  $\rho$  (SI unit: kg/m<sup>3</sup>) of the surrounding fluid. This can either be taken **From material** or **User defined**, in which case the default is 1 kg/m<sup>3</sup>.

## ACOUSTOPHORETIC FORCE

Select an option from the **Specify** list: **Particle bulk modulus** (the default), **Particle compressibility**, or **Compressional speed of sound**. Then, if required change the defaults and enter a value or expression the available fields as follows:

- **Particle bulk modulus**  $K_p$  (SI unit: Pa). The default is 2.2 GPa ( $2.2 \times 10^9$ ).
- **Particle compressibility**  $\beta_p$  (SI unit: 1/Pa). The default is  $1/(2.2 \times 10^9)$  1/Pa.
- **Compressional speed of sound**  $c_p$  (SI unit: m/s). The default is 3000 m/s.

## ADVANCED SETTINGS

Select the **Use piecewise polynomial recovery on field** check box to smooth the radiation pressure using piecewise polynomial recovery. This can give a much more accurate representation of the radiation pressure because it uses information on adjacent mesh elements to reconstruct the field. If a coarse mesh is used to compute the field then this option can be especially useful.



- [Studies and Solvers](#) and [Frequency Domain](#) in the *COMSOL Multiphysics Reference Manual*
- [Additional Forces](#) in the theory section.

## *Electric Force*



The **Electric Force** node is only available when **Newtonian** or **Newtonian, first order** is selected as the **Formulation** on the **Settings** window for [The Particle Tracing for Fluid Flow Interface](#).

Use the **Electric Force** node to exert an electric force to the particles. The force is specified via an electric potential or the electric field. For cases where the field was computed in the frequency domain, the force can be computed by multiplying the field by the phase angle. Additionally, piecewise polynomial recovery can be used which can give a more accurate representation of the specified electric field.

The settings for this feature node are the same as for the [Electric Force](#) node described for [The Charged Particle Tracing Interface](#).



[Additional Forces](#) in the theory section.

---

### *Magnetic Force*

---



The **Magnetic Force** node is only available when **Newtonian** or **Newtonian, first order** is selected as the **Formulation** on the **Settings** window for [The Particle Tracing for Fluid Flow Interface](#).

---

Use the **Magnetic Force** node to exert a magnetic force to the particles. A magnetic force alone does not work on the particles, so in the absence of any other external forces, the particle retains its original energy. The force is specified via a magnetic flux density.

The settings for this feature node are the same as for the [Magnetic Force](#) node described for [The Charged Particle Tracing Interface](#).



[Additional Forces](#) in the theory section.

---

### *Dielectrophoretic Force*

---



The **Dielectrophoretic Force** node is only available when **Newtonian** or **Newtonian, first order** is selected as the **Formulation** on the **Settings** window for [The Particle Tracing for Fluid Flow Interface](#).

---

Use the **Dielectrophoretic Force** node to exert a dielectrophoretic force to the particles. The force is specified via an electric potential or the electric field. The force has a

different meaning depending on whether the source electric field is computed in a **Stationary** or **Frequency Domain** study.

Additionally, piecewise polynomial recovery can be used which can give a more accurate representation of the specified electric field. The influence of the dielectrophoretic force on the particles depends on the difference in permittivity between the particles and the fluid. When the particle relative permittivity is greater than the relative permittivity of the fluid, the particles are attracted to regions where the absolute electric field is strong. When the particle relative permittivity is less than the relative permittivity of the fluid, the particles are attracted to regions where the absolute electric field is weak.

The **Shell** subnode is available from the context menu (right-click the parent node) or from the **Physics** toolbar, **Attributes** menu. Use it to model dielectrophoresis of particles with thin dielectric shells. If no **Shell** subnodes are added, the particle is treated as a homogeneous sphere of uniform permittivity and conductivity for the purpose of computing the dielectrophoretic force.

#### **DIELECTROPHORETIC FORCE**

Select an option from the **Specify force using** list: **Electric field** (the default) or **Electric potential**.

- For **Electric field** enter values or expressions in the table for the **Electric field**  $\mathbf{E}$  (SI unit: V/m) based on space dimension. If the electric field is computed by another physics interface then it can be selected from the list.
- For **Electric potential** enter a value or expression for **Electric potential**  $V$  (SI unit: V). If the electric potential is computed by another physics interface then it can be selected from the list.

#### **PARTICLE PROPERTIES**

Enter a value or expression for the **Particle relative permittivity**  $\epsilon_{r,p}$  (dimensionless). The default value is 1. The value must be greater than or equal to one. Then enter a value or expression for the **Particle electrical conductivity**  $\sigma_p$  (SI unit: S/m). The default value is 0.

#### **FLUID PROPERTIES**

By default the **Relative permittivity**  $\epsilon_r$  (dimensionless) is taken **From material**. Select **User defined** from the list to enter a value or expression.

By default the **Electrical conductivity**  $\sigma$  (SI unit: S/m) is taken **From material**. Select **User defined** from the list to enter a value or expression.

### ADVANCED SETTINGS

Select the **Use piecewise polynomial recovery on field** check box to smooth the electric field using piecewise polynomial recovery. This can give a much more accurate representation of the electric field as it uses information on adjacent mesh elements to reconstruct the field. If a coarse mesh is used to compute the field then this option can be especially useful.

Select an option from the **Angular frequency** list—**From solution** or **User defined**. For **From solution** the dielectrophoretic force is computed using the angular frequency of the electric field computed in a previous study step. For **User defined** enter the **Angular frequency**  $\omega$  (SI unit: rad/s). The default value is 0.

Use the **Particles to affect** list to apply the force to specific particles. The available settings are the same as for the [Force](#) node.



- [Studies and Solvers](#) in the *COMSOL Multiphysics Reference Manual*
  - [Additional Forces](#) in the theory section.
- 

### Shell

---

The **Shell** subnode is available from the context menu (right-click the [Dielectrophoretic Force](#) parent node) or from the **Physics** toolbar, **Attributes** menu. Add **Shell** subnodes to a **Dielectrophoretic Force** node to compute an equivalent Clausius-Mossotti factor using the electrical properties of the interior of the particle and one or more thin layers on its surface.

If multiple **Shell** subnodes are present, their order in the Model Builder determines how the nested shells are modeled, with the first subnode constituting the innermost layer.

### SHELL PROPERTIES

Enter the **Shell thickness**  $t_s$  (SI unit: m). The default is  $1 \times 10^{-7}$  m. Enter a value for the **Shell relative permittivity**  $\epsilon_{r,s}$  (dimensionless). The default is 1. Enter a **Shell electrical conductivity**  $\sigma_s$  (SI unit: S/m). The default is 1 S/m.



[Dielectrophoretic Force](#) in the theory section.

---

## Magnetophoretic Force

---



The **Magnetophoretic Force** node is only available when **Newtonian** or **Newtonian, first order** is selected as the **Formulation** on the **Settings** window for [The Particle Tracing for Fluid Flow Interface](#).

---

Use the **Magnetophoretic Force** node to include forces on particles due to a difference in permeability between the particles and background fluid. The force is specified via a magnetic field and the permeability of the particle and the fluid.

Additionally, piecewise polynomial recovery can be used, which can give a more accurate representation of the specified magnetic field.

### MAGNETOPHORETIC FORCE

Enter a value or expression for the **Magnetic field  $\mathbf{H}$**  (SI unit: A/m) based on space dimension, or if the field is computed by another physics interface then it can be selected from the list. Unlike the [Dielectrophoretic Force](#) the **Magnetophoretic Force** always treats the model particles as homogeneous spheres.



The magnetophoretic force is a function of the spatial derivatives of the magnetic field components. For formulations based on the magnetic scalar potential  $\mathbf{A}$ , these derivatives of  $\mathbf{H}$  may not be available. See [Additional Forces](#) in the theory section for more details.

---

### PARTICLE PROPERTIES

Enter a value or expression for the **Particle relative permeability  $\mu_{r,p}$**  (dimensionless). The value must be greater than or equal to one.

### FLUID PROPERTIES

Enter a value or expression for the **Fluid relative permeability  $\mu_{r,f}$**  (dimensionless). The value must be greater than or equal to one.

### ADVANCED SETTINGS

Select the **Use piecewise polynomial recovery on field** check box to smooth the magnetic field using piecewise polynomial recovery. This can give a much more accurate representation of the magnetic field as it uses information on adjacent mesh elements to reconstruct the field. If a coarse mesh is used to compute the field then this option can be especially useful.



Use the **Particles to affect** list to apply the force to specific particles. The available settings are the same as for the [Force](#) node.



[Additional Forces](#) in the theory section.

---

### *Thermophoretic Force*

---



The **Thermophoretic Force** node is only available when **Newtonian** or **Newtonian, first order** is selected as the **Formulation** on the **Settings** window for [The Particle Tracing for Fluid Flow Interface](#).

Use the **Thermophoretic Force** node to include forces on particles due to a temperature gradient in the background fluid. The force is specified via the background temperature and other properties of the fluid.

Additionally, piecewise polynomial recovery can be used which can give a more accurate representation of the specified temperature.

#### **THERMOPHORETIC FORCE**

If the **Include rarefaction effects** check box is selected in the physics interface **Particle Release and Propagation** section, select a **Thermophoretic force model**—**Epstein**, **Waldmann, Talbot** (the default), or **Linearized BGK**. Use these settings to improve the accuracy of the thermophoretic force calculation in fluids with high Knudsen numbers. When the **Include rarefaction effects** check box is cleared, the fluid is treated as a continuum and only the Epstein model is used.

Enter a value or expression for the **Temperature**  $T$  (SI unit: K), or if the temperature is computed by another physics it can be selected from the list. The default is 293.15 K.

The following fluid parameters all take default values **From material** or can be computed from another physics interface. Or select **User defined** to enter a value or expression for the:

- **Dynamic viscosity**  $\mu$  (SI unit: Pa·s). The default is 2E-5 Pa·s.
- **Density**  $\rho$  (SI unit: kg/m<sup>3</sup>). The default is 1 kg/m<sup>3</sup>.

- **Thermal Conductivity**  $k$  (SI unit: W/(m·K)). For **User defined** select **Isotropic**, **Diagonal**, **Symmetric**, or **Anisotropic** and then enter values in the field or matrix.
- Heat capacity at constant pressure  $C_p$  (SI unit: J/(kg·K)). The default is 1000.

If the **Waldmann**, **Talbot**, or **Linearized BGK** model is used, enter the **Background gas molar mass**  $M_g$  (SI unit: kg/mol). The default is 0.04 [kg/mol].

If the **Waldmann**, **Talbot**, or **Linearized BGK** model is used, select an option from the **Mean free path calculation** list: **Ideal gas**, **hard sphere collisions** or **User-defined correction**. For **User-defined correction** enter a **Mean free path correction**  $\lambda'/\lambda$  by which the mean free path differs from the value for hard spheres in an ideal gas.

If the **Linearized BGK** model is used, enter an **Energy accommodation coefficient**  $\alpha_E$  (dimensionless) and a **Momentum accommodation coefficient**  $\alpha_M$  (dimensionless). The default values are 1.

#### PARTICLE PROPERTIES

Enter a value or expression for the:

- **Particle thermal conductivity**  $k_p$  (SI unit: W/(m·K)). The default is 1 W/(m·K).
- **Thermophoretic correction factor**  $C_s$  (dimensionless). The default is 1.17.
- If the **Talbot** model is used, enter two additional dimensionless thermophoretic correction factors  $C_m$  and  $C_t$ . The default values are 1.146 and 2.2, respectively.

#### ADVANCED SETTINGS

Select the **Use piecewise polynomial recovery on field** check box to smooth the temperature using piecewise polynomial recovery. This can give a much more accurate representation of the temperature as it uses information on adjacent mesh elements to reconstruct the field. If a coarse mesh is used to compute the field then this option can be especially useful.

Use the **Particles to affect** list to apply the force to specific particles. The available settings are the same as for the [Force](#) node.



[Additional Forces](#) in the theory section.

---

## Erosion

---

The **Erosion** subnode is available from the context menu (right-click the **Wall** parent node) or from the **Physics** toolbar, **Attributes** menu. Use it to calculate the rate of erosive wear or the total mass removed due to the impact of particles on a boundary.

- When **Specify release times** is selected from the **Particle release specification** list in the physics interface **Particle Release and Propagation** section, the **Erosion** node calculates the total mass lost per unit area (SI unit:  $\text{kg}/\text{m}^2$ ).
- If **Specify mass flow rate** is selected from the **Particle release specification** list, the **Erosion** node calculates the rate of erosive wear (SI unit:  $\text{kg}/(\text{m}^2\text{s})$ ).

### EROSION MODEL

Select an **Erosion model**: **Expression**, **Finnie** (default), **E/CRC**, **Oka**, or **DNV**. Then enter the following settings:

#### Expression

- If **Specify release times** is selected from the **Particle release specification** list in the physics interface **Particle Release and Propagation** section, enter the **Mass removed by particle**  $\Delta m_i$  (SI unit: kg). The default is  $1\text{e}-15$  kg.
- If **Specify mass flow rate** is selected from the **Particle release specification** list, enter the **Erosive wear of particle**  $E_{M,i}$  (SI unit: kg/s). The default is  $1\text{e}-15$  kg/s.

#### Finnie

- **Fraction of particles cutting in an idealized manner**  $c_i$  (dimensionless). The default is 0.1.
- **Ratio of normal and tangential forces**  $K$  (dimensionless). The default is 2.
- **Surface hardness**  $H_V$  (SI unit: N/m). The default is  $2\text{e}9$  Pa.
- **Mass density of surface**  $\rho$  (SI unit:  $\text{kg}/\text{m}^3$ ). The default is  $7.5\text{e}3[\text{kg}/\text{m}^3]$ .

#### E/CRC

- **E/CRC model coefficient**  $C$  (dimensionless).
- **Brinell hardness of surface material**  $BH$  (dimensionless). The default is 200.
- **Particle shape coefficient**  $F_s$  (dimensionless). The default is 0.2.
- **E/CRC model exponent**  $n$  (dimensionless). The default is 2.41.

#### Oka

- **Oka model coefficient**  $K$  (SI unit:  $\text{m}^3/\text{kg}$ ). The default is  $65[\text{mm}^3/\text{kg}]$ .
- **Reference diameter**  $d_{\text{ref}}$  (SI unit: m). The default is  $326[\mu\text{m}]$ .

- **Reference velocity**  $v_{\text{ref}}$  (SI unit: m/s). The default is 104 [m/s].
- Each of the following dimensionless **Oka model coefficient**:
  - $k_1$  (default is -0.12) and  $k_2$  (default is 0.19).
  - $q_1$  default is 0.14) and  $q_2$  (default is -0.94).
  - $s_1$  (default is 0.71) and  $s_2$  (default is 2.4).
- **Surface hardness**  $H_V$  (SI unit: N/m<sup>2</sup>). The default is 2e9 [N/m<sup>2</sup>].
- **Mass density of surface**  $\rho$  (SI unit: kg/m<sup>3</sup>). The default is 7.5e3 [kg/m<sup>3</sup>].

#### DNV

- **DNV model coefficient**  $K$  (dimensionless). The default is 2e-9.
- **DNV mode exponent**  $n$  (dimensionless). The default is 2.6.

If **Specify release times** is selected from the **Particle release specification** list in the physics interface **Particle Release and Propagation** section, it is possible to multiply the eroded mass by another scale factor. Select an option from the **Number multiplication factor specification** list—**From physics** (the default) or **User defined**.

For **From physics** the extra scale factor is usually equal to 1. However, if the **Enable macroparticles** check box is selected in the physics interface **Additional Variables** section, the scale factor is instead taken from the built-in auxiliary dependent variable for the particle number multiplication factor, which can be initialized in the Settings windows for most particle release features.

For **User defined**, specify the extra scale factor directly. Enter a value or expression for the **Number multiplication factor**  $n_n$  (dimensionless). The default is 1.



*Pipe Erosion due to Contaminant Particles*: Application Library path  
**Particle\_Tracing\_Module/Fluid\_Flow/pipe\_elbow\_erosion**

---

### *Mass Deposition*

If **Specify release times** is selected from the **Particle release specification** list in the physics interface **Particle Release and Propagation** section, the **Mass Deposition** subnode is available from the context menu (right-click the [Wall](#) parent node) or from the **Physics** toolbar, **Attributes** menu. The **Mass Deposition** subnode uses an accumulated variable on each boundary element in the selection of the parent node to calculate the mass of particles deposited on the surface, expressed in mass per unit area.

## *Boundary Load*

---

If **Specify mass flow rate** is selected from the **Particle release specification** list in the physics interface **Particle Release and Propagation** section, the **Boundary Load** subnode is available from the context menu (right-click the [Wall](#) parent node) or from the **Physics** toolbar, **Attributes** menu. The **Boundary Load** subnode uses accumulated variables on each boundary element in the selection of the parent node to calculate the force due to the impact of particles.

### **BOUNDARY LOAD**

Select a **Type**—**Pressure** or **Force per unit area**. For **Force per unit area** accumulated variables are defined for components of the boundary load vector. For **Pressure** one accumulated variable is defined for the normal force per unit area in each boundary element.

## *Mass Flux*

---

If **Specify mass flow rate** is selected from the **Particle release specification** list in the physics interface **Particle Release and Propagation** section, the **Mass Flux** subnode is available from the context menu (right-click the [Wall](#) parent node) or from the **Physics** toolbar, **Attributes** menu. The **Mass Flux** subnode uses accumulated variables on each boundary element in the selection of the parent node to calculate the mass flux of particles at a boundary.

### **MASS FLUX**

Select a **Type**—**Mass flux** or **Normal mass flux**. For **Mass flux** accumulated variables are defined for components of the mass flux vector. For **Normal mass flux** one accumulated variable is defined for the normal mass flux in each boundary element.

## *Volume Force Calculation*

---

Use the **Volume Force Calculation** node if the particles exert a significant force on the surrounding fluid. This node defines dependent variables for the components of the volume force; the net volume force is equal in magnitude and opposite in direction to the total

on the particles.



To compute the volume force exerted by the particles on the fluid, at least one [Drag Force](#) node must be present.



The **Volume Force Calculation** node computes the volume force in the same way as the [Fluid-Particle Interaction](#) Multiphysics node but does not automatically include the accumulated variables as source terms when computing the fluid velocity and pressure. For modeling bidirectionally coupled fluid-particle interactions, consider using [The Fluid-Particle Interaction Interface](#) instead of the **Volume Force Calculation** node.

For more information about the calculation of the volume force, see [Volume Force Calculation](#) in the section [Theory for the Fluid-Particle Interaction Interface](#).

#### VOLUME FORCE CALCULATION

This section is available if **Specify release times** is selected from the **Particle release specification** list in the physics interface **Particle Release and Propagation** section. Select an option from the **Force multiplication factor specification** list—**From physics** (the default) or **User defined**.

- If **From physics** is selected and the **Enable macroparticles** check box is cleared in the physics interface **Additional Variables** section, the multiplication factor is 1.
- If **From physics** is selected and the **Enable macroparticles** check box is selected in the physics interface **Additional Variables** section, the multiplication factor is based on the auxiliary dependent variable for the multiplication factor, which is typically specified in release feature settings. If the physics interface has name `fpt`, this variable has name `fpt.nn`.
- If **User defined** is selected, enter a value or expression for the **Force multiplication factor**  $n$  (dimensionless). The default is 1.

The **Force multiplication factor** can be used to represent each particle as a group of  $n$  particles that follow the same trajectory. This means that the magnitude of the contribution to the volume force by each particle is multiplied by  $n$ .



If **Specify mass flow rate** is selected from the **Particle release specification** list in the physics interface **Particle Release and Propagation** section, the **Force multiplication factor**  $n$  cannot be specified because the number of particles represented by each model particle is instead controlled by the **Mass flow rate**, which is specified in the settings for release features such as the [Release](#) and [Inlet](#) nodes.

---

### *Convective Heat Losses*

The **Convective Heat Losses** node is available when the **Compute particle temperature** check box is selected in the physics interface **Additional Variables** section. Use the **Convective Heat Losses** node to model the heating or cooling of a particle due to convective heat exchange with the surrounding fluid. The temperature within the particle is assumed to be uniform; that is, heat transfer by conduction within the particle takes place on a much shorter time scale than heat transfer by convection at the surface.

#### **CONVECTIVE HEAT LOSSES**

Enter a **Heat transfer coefficient**  $h$  (SI unit:  $\text{W}/(\text{m}^2 \cdot \text{K})$ ). The default is 0. Enter a value or expression for the **Temperature**,  $T$  (SI unit: K), or if the temperature is computed by another physics it can be selected from the list. The default is 293.15 K.



[Computing Particle Temperature](#) in the theory section.

---

### *Radiative Heat Losses*

The **Radiative Heat Losses** node is available when the **Compute particle temperature** check box is selected in the physics interface **Additional Variables** section. Use the **Radiative Heat Losses** node to model the heating or cooling of a particle due to radiative heat exchange with its surroundings according to the Stefan-Boltzmann law for diffuse gray surfaces.

## RADIATIVE HEAT LOSSES

Enter a **Particle emissivity**  $\varepsilon_p$  (dimensionless). The default is 1. Enter a value or expression for the **Temperature**,  $T$  (SI unit: K), or if the temperature is computed by another physics it can be selected from the list. The default is 293.15 K.



[Computing Particle Temperature](#) in the theory section.

---

## *Heat Source*

---

The **Heat Source** node is available when the **Compute particle temperature** check box is selected in the physics interface **Additional Variables** section. Use the **Heat Source** node to apply a source term that affects the particle temperature.

## HEAT SOURCE

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



[Computing Particle Temperature](#) in the theory section.

---

## *Droplet Breakup*

---

Use the **Droplet Breakup** node to model the breakup of liquid droplets or parcels of droplets.

To use the **Droplet Breakup** feature, the following prerequisites must be met:

- **Newtonian** or **Newtonian, first order** must be selected from the **Formulation** list in the physics interface **Particle Release and Propagation** section.
- The **Compute particle mass** check box must be selected in the physics interface **Additional Variables** section.
- **Liquid droplets/bubbles** must be selected from the **Particle type** list in the settings window for either the [Particle Properties](#) node or at least one instance of the [Override Properties](#) feature.
- At least one instance of the [Drag Force](#) feature is needed. The valid selection for the **Droplet Breakup** feature comprises all domains where drag forces are defined.



The **Kelvin-Helmholtz Breakup Model** and **Rayleigh-Taylor Breakup Model** subnodes are available from the context menu (right-click the **Droplet Breakup** parent node) or from the **Physics** toolbar, **Attributes** menu. Only a single instance of each feature can be created. The **Kelvin-Helmholtz Breakup Model** is created by default.



To model droplet breakup, unreleased secondary particles must be available. To ensure that all child droplets are released properly, a sufficiently large value must be specified for the **Maximum number of secondary particles** in the physics interface **Particle Release and Propagation** section.

---

### *Kelvin-Helmholtz Breakup Model*

---

Use the **Kelvin-Helmholtz breakup model** to account for the effect of Kelvin-Helmholtz instability on the breakup of liquid droplets. This causes small child droplets to be stripped away from larger droplets over time.

#### **BREAKUP MODEL**

Enter a value or expression for the **Kelvin-Helmholtz model coefficient**  $B_{KH}$  (dimensionless). The default value is 10.

If the **Enable macroparticles** check box is selected in the physics interface **Additional Variables** section, enter a value or expression for the **Maximum number of particles released per breakup**  $N_{max}$  (dimensionless). The default is 10. This prevents an inordinate amount of child droplets from being released. If more child droplets than the maximum specified number would be released, then the multiplication factor of each particle is adjusted to compensate for the limited number of model particles.

#### **GAS PROPERTIES**

The **Density**  $\rho$  (SI unit:  $\text{kg/m}^3$ ) is taken **From material**. For **User defined** enter another value or expression. The default is  $1 [\text{kg/m}^3]$ .

Enter the components of the **Velocity field**  $\mathbf{u}$  (SI unit:  $\text{m/s}$ ) based on space dimension. If another physics interface is present which computes the velocity field then this can be selected from the list.

## *Rayleigh-Taylor Breakup Model*

---

Use the **Rayleigh-Taylor breakup model** to account for the effect of Rayleigh-Taylor instability on the breakup of liquid droplets. Rayleigh-Taylor breakup causes the parent droplet to be completely broken up into smaller child droplets, and is usually associated with droplets undergoing large accelerations.

### **BREAKUP MODEL**

Specify the **Rayleigh-Taylor coefficient**  $C_{RT}$  (dimensionless). The default value is 5.

If the **Enable macroparticles** check box is selected in the physics interface **Additional Variables** section, enter a value or expression for the **Maximum number of particles released per breakup**  $N_{max}$  (dimensionless). The default is 10. This prevents an inordinate amount of child droplets from being released. If more child droplets than the maximum specified number would be released, then the multiplication factor of each particle is adjusted to compensate for the limited number of model particles.



### [Droplet Breakup Theory](#)

---

## *Nozzle*

---

Use the **Nozzle** feature to release a spray of liquid droplets.

To use the **Nozzle** feature, the following prerequisites must be met:

- **Newtonian** or **Newtonian, first order** must be selected from the **Formulation** list in the physics interface **Particle Release and Propagation** section.
- **Specify release times** must be selected from the **Particle release specification** list in the physics interface **Particle Release and Propagation** section.
- The **Compute particle mass** and **Enable macroparticles** check boxes must be selected in the physics interface **Additional Variables** section.
- **Liquid droplets/bubbles** must be selected from the **Particle type** list in the settings window for either the [Particle Properties](#) node or at least one instance of the [Override Properties](#) feature.

The **Nozzle** feature is supported on two geometric entity levels. If it is added as a global feature (similar to the [Release from Grid](#) feature), the initial position of the injected droplets can be specified by entering a list of coordinates. If it is added as a 0D feature

(similar to the [Release from Point](#) feature), the initial positions of the injected droplets correspond to the selected points in the geometry.

Go to [Release](#) for information about the following sections: **Inherit Properties**, **Initial Value of Auxiliary Dependent Variables**, and **Advanced Settings**.

### RELEASE TIMES

Enter values for the **Release times** (SI unit: s). The default expression is  $0 \text{ } 1 \text{ [ms]}$ . At least two release times must be specified, and the release times must increase monotonically. This is because the time intervals between successive release times are used to assign appropriate number multiplication factors to the released particles so that the mass flow rate is constant in a finite time interval.

### INITIAL VELOCITY

Select an option from the Velocity specification list—**From mass flow rate** (the default), **From change in pressure**, **From injection pressure**, or **User defined**.


For **From mass flow rate** enter a value or expression for the **Mass flow rate**  $\dot{m}$  (SI unit: kg/s). The default is  $1 \text{e-}3 \text{ [kg/s]}$ .

For **From change in pressure** enter a value or expression for the **Change in pressure**  $\Delta p$  (SI unit: Pa). The default value is  $1 \text{ [MPa]}$ . Then enter a value or expression for the **Discharge coefficient**  $c_d$  (dimensionless). The default value is  $0.7$ .

For **From injection pressure** enter a value or expression for the **Injection pressure**  $p_{inj}$  (SI unit: Pa). The default value is  $1 \text{ [MPa]}$ . Then enter a value or expression for the **Discharge coefficient**  $c_d$  (dimensionless). The default value is  $0.7$ .

For **User defined** enter a value or expression for the **Initial droplet velocity magnitude**  $U_0$  (SI unit: m/s). The default value is  $100 \text{ m/s}$ .

### INJECTION PARAMETERS

Specify the initial droplet positions based on space dimension ( $q_{x,0}$ ,  $q_{y,0}$ , and  $q_{z,0}$  for 3D components) or click the **Range** button () to select and define a range of specific coordinates.

Select an option from the **Injector Location** list: **All combinations** (the default) or **Specified combinations**. If **Specified combinations** is selected, the number of initial coordinates entered for each space dimension must be equal, and the total number of release positions is equal to the length of one of the lists of initial coordinates. If **All combinations** is selected, the total number of release positions is equal to the product of the lengths of each list of initial coordinates.

For example, suppose a 2D component includes a **Nozzle** 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 initial positions will be released, including every possible combination of the initial  $x$ - and  $y$ -coordinates. If **Specified combinations** is selected, particles will only be released at 4 initial positions (0,2), (1,4), (2,6), and (3,8).

The position for any particles with initial coordinates outside the geometry are set to NaN, so the particles do not appear when plotted during postprocessing.

If the **Nozzle** feature is added as a 0D feature (similar to the [Release from Point](#) feature), the **Injector location** list and the edit fields for the initial coordinates are not shown, and the initial coordinates are instead specified by selecting points in the geometry.

Enter coordinates for the **Injector axis  $\mathbf{r}$**  (dimensionless) based on space dimension. By default the injector axis points in the positive  $x$ -direction.

Enter a value or expression for the **Nozzle exit radius  $r_0$**  (SI unit: m), which determines the initial size of released droplets.

Enter a value or expression for the **Number of parcels per release  $N_0$**  (dimensionless).



The **Number of parcels per release** does not actually restrict the number of droplets or the total mass of fluid that can be released by the **Nozzle** feature; it only restricts the number of model particles that can be released. Each model particle represents a parcel representing an arbitrary number of droplets. The number of droplets per model particle is always available as an auxiliary dependent variable because the **Enable macroparticles** check box must be selected in the physics interface **Additional Variables** section in order to use this feature.

### MAXIMUM CONE ANGLE

Select an option from the **Maximum cone angle specification** list—**User defined** (the default) or **From Kelvin-Helmholtz instability model**.

If **User defined** is selected, enter a value or expression for the **Maximum cone angle  $\Theta$**  (SI unit: rad). The default value is 10[deg].

If **From Kelvin-Helmholtz instability model** is selected, specify the **Spray angle constant  $A$**  (dimensionless). The default value is **0.188**.



## Nozzle Theory

---

### *Nozzle Domain*

---

The **Nozzle Domain** feature is automatically added as a subnode to the **Nozzle** feature. It can never be added or removed. Use the **Nozzle Domain** feature to specify which fluid domain the droplets are being injected into, and to specify material properties in this domain.

The **Density  $\rho$**  (SI unit:  $\text{kg/m}^3$ ) is taken **From material**. For **User defined** enter another value or expression.



Enter a value or expression for the **Pressure  $p$**  (SI unit: Pa). If another physics interface is present which computes the pressure then this can be selected from the list.



To use the **Nozzle** release feature, at least one domain must be in the selection of the **Nozzle Domain** subnode.

---

# The Droplet Sprays in Fluid Flow Interface

The **Droplet Sprays in Fluid Flow** interface (  ), found under the **Fluid Flow>Particle Tracing** branch (  ) when adding a physics interface, creates an instance of the **Particle Tracing for Fluid Flow** interface with specialized default settings and features that are often used to model sprays of liquid droplets.

All available physics features and settings are the same as for [The Particle Tracing for Fluid Flow Interface](#), with the exception that the following defaults are used:

- The **Compute particle mass** check box is selected in the physics interface **Additional Variables** section,
- **Liquid droplets/bubbles** is selected from the **Particle type** list in the settings window for the default [Particle Properties](#) node,
- The [Drag Force](#) node is added to all domains by default, and
- The [Droplet Breakup](#) node is added to all domains by default. The [Kelvin-Helmholtz Breakup Model](#) node is added as a default subnode to the **Droplet Breakup** node.

# Theory for the Particle Tracing for Fluid Flow Interface

The Particle Tracing for Fluid Flow Interface theory is described in this section:

- [Particle Motion in a Fluid](#)
- [Particle Motion in a Turbulent Flow](#)
- [Particle Motion in a Shear Flow](#)
- [Additional Forces](#)
- [References for the Particle Tracing for Fluid Flow Interface](#)

## *Particle Motion in a Fluid*

---

The particle momentum comes from Newton's second law, which states that the net force on a particle is equal to its time rate of change of its linear momentum in an inertial reference frame:

$$\frac{d}{dt}(m_p \mathbf{v}) = \mathbf{F}_D + \mathbf{F}_g + \mathbf{F}_{\text{ext}}$$

When the particle mass is being computed, such that accretion or evaporation can take place, the mass is moved outside the time derivative to prevent non-physical acceleration of the particles:

$$m_p \frac{d\mathbf{v}}{dt} = \mathbf{F}_D + \mathbf{F}_g + \mathbf{F}_{\text{ext}}$$

The assumption is that any mass lost by the particles continues to move with the particle velocity and doesn't cause the particle to decelerate.

### **DRAG FORCE**

In the above equation,  $\mathbf{F}_D$  is the [Drag Force](#) is defined as:

$$\mathbf{F}_D = \left(\frac{1}{\tau_p}\right) m_p (\mathbf{u} - \mathbf{v})$$

where

- $m_p$  is the particle mass (SI unit: kg)

- $\tau_p$  is the particle velocity response time (SI unit: s)
- $\mathbf{v}$  is the velocity of the particle (SI unit: m/s)
- $\mathbf{u}$  is the fluid velocity (SI unit: m/s)
- $\mathbf{F}_g$  is the gravitational force vector (SI unit: N), and
- $\mathbf{F}_{\text{ext}}$  is any other external force (SI unit: N).

A large number of expressions for the particle response time are available; selecting an appropriate drag law requires knowledge of the relative Reynolds number  $\text{Re}_r$  (dimensionless) of particles in the flow. The relative Reynolds number is given by the expression

$$\text{Re}_r = \frac{\rho \|\mathbf{u} - \mathbf{v}\| d_p}{\mu}$$

Many drag laws and the relative Reynolds numbers at which they are applicable are given in [Ref. 4](#).

#### *Stokes*

The particle velocity response time for spherical particles in a laminar flow is defined as:

$$\tau_p = \frac{\rho_p d_p^2}{18\mu}$$

where  $\mu$  is the fluid viscosity (SI unit: Pa s),  $\rho_p$  is the particle density (SI unit: kg/m<sup>3</sup>), and  $d_p$  is the particle diameter (SI unit: m). This is frequently known as **Stokes** drag law. The Stokes drag law is applicable for creeping flows,  $\text{Re}_r \ll 1$ .

#### *Schiller-Naumann*

When the drag law is set to **Schiller-Naumann**, the particle velocity response time is redefined as:

$$\tau_p = \frac{4\rho_p d_p^2}{3\mu C_D \text{Re}_r}$$

where:

$$C_D = \frac{24}{\text{Re}_r} (1 + 0.15 \text{Re}_r^{0.687})$$

The Schiller-Naumann drag law is applicable at moderate relative Reynolds numbers,  $\text{Re}_r < 800$ .



### Haider-Levenspiel

When the drag law is set to **Haider-Levenspiel**, the drag coefficient is given by

$$C_D = \frac{24}{\text{Re}_r} \left( 1 + A(S_p) \text{Re}_r^{B(S_p)} \right) + \frac{C(S_p)}{1 + D(S_p)/\text{Re}_r}$$

where  $A$ ,  $B$ ,  $C$  and  $D$  are empirical correlations of the particle sphericity. The sphericity is defined as the ratio of the surface area of a volume equivalent sphere to the surface area of the considered non-spherical particle

$$0 < S_p = \frac{A_{\text{sphere}}}{A_{\text{particle}}} \leq 1$$

The correlation coefficients are given by

$$A(S_p) = \exp(2.3288 - 6.4581S_p + 2.4486S_p^2)$$

$$B(S_p) = 0.0964 + 0.5565S_p$$

$$C(S_p) = \exp(4.905 - 13.8944S_p + 18.4222S_p^2 - 10.2599S_p^3)$$

$$D(S_p) = \exp(1.4681 + 12.2584S_p - 20.7322S_p^2 + 15.8855S_p^3)$$

The diameter used in the Reynolds number is that of the volume equivalent sphere.

### Oseen Correction

When the drag law is set to **Oseen correction**, the drag coefficient is given by

$$C_D = \frac{24}{\text{Re}_r} \left( 1 + \frac{3}{16} \text{Re}_r \right)$$

Like **Stokes** drag law, the Oseen correction is applicable at low relative Reynolds numbers, typically  $\text{Re}_r < 0.1$ . The **Oseen correction** is determined by simplifying, rather than neglecting, the inertia term in the Navier-Stokes equation, resulting in drag coefficients that are slightly greater than those based on the Stokes drag law.

### Hadamard-Rybczynski

The **Hadamard-Rybczynski** drag law is based on an analytical solution for creeping flow past a spherical liquid drop or gas bubble. The interface between the two fluids is assumed to be completely free of surface-active contaminants. The drag coefficient is given by the expression

$$C_D = \frac{8}{\text{Re}_r} \left( \frac{2 + 3\kappa}{1 + \kappa} \right)$$

where  $\kappa = \mu_p/\mu$  is the ratio of the particle viscosity and fluid viscosity. For arbitrarily large values of  $\mu_p$  this expression asymptotically approaches the **Stokes** law.

The Hadamard-Rybczynski drag law is only applicable if the droplets and the surrounding fluid are extremely pure and free from contaminants. If the fluids are not sufficiently pure, the drag force on the bubble or droplet is more likely to be accurately determined by the Stokes drag law.

When the drag law is set to **Standard drag correlations**, the drag coefficient is defined as a piecewise function of the relative Reynolds number:

RANGE	CORRELATION
$Re_r \leq 0.01$	$C_D = \frac{24}{Re_r} \left(1 + \frac{3}{16} Re_r\right)$
$0.01 < Re_r \leq 20$	$C_D = \frac{24}{Re_r} (1 + 0.1315 Re_r^{0.82 - 0.05w})$
$20 < Re_r \leq 260$	$C_D = \frac{24}{Re_r} (1 + 0.1935 Re_r^{0.6305})$
$260 < Re_r \leq 1500$	$\log C_D = 1.6435 - 1.1242w + 0.1558w^2$
$1500 < Re_r \leq 1.2 \times 10^4$	$\log C_D = -2.4571 + 2.5558w - 0.9295w^2 + 0.1049w^3$
$1.2 \times 10^4 < Re_r \leq 4.4 \times 10^4$	$\log C_D = -1.9181 + 0.6370w - 0.0636w^2$
$4.4 \times 10^4 < Re_r \leq 3.38 \times 10^5$	$\log C_D = -4.3390 + 1.5809w - 0.1546w^2$
$3.38 \times 10^5 < Re_r \leq 4 \times 10^5$	$C_D = 29.78 - 5.3w$
$4 \times 10^5 < Re_r \leq 1 \times 10^6$	$C_D = 0.1w - 0.49$
$1 \times 10^6 < Re_r$	$C_D = 0.19 - \frac{8 \times 10^4}{Re_r}$

where  $w = \log Re_r$  and the base 10 logarithm has been used.

The **Standard drag correlations** can be used when the relative Reynolds number is expected to change by several orders of magnitude during a simulation. It is also applicable at significantly higher relative Reynolds numbers than the **Schiller-Naumann** drag law.

The drag coefficient as a function of the relative Reynolds number is shown in Figure 5-1.

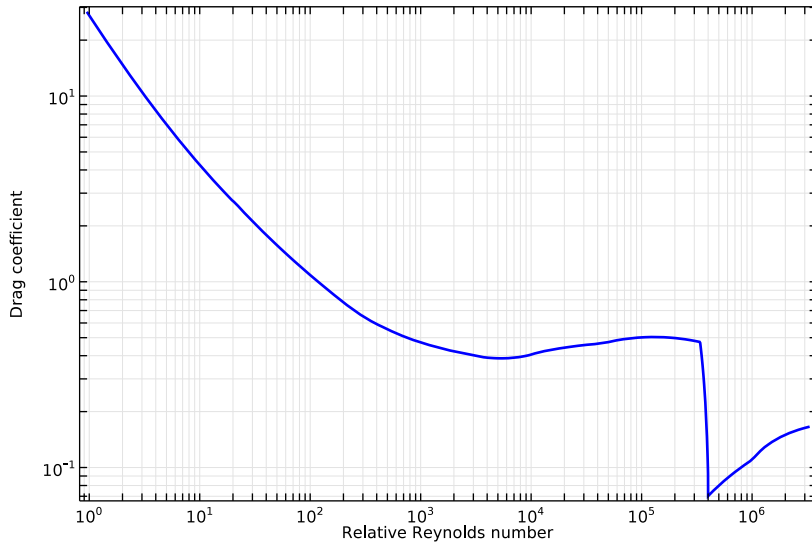


Figure 5-1: The piecewise function used to compute the drag coefficient when Standard Drag Correlations are used.

#### DRAG FORCE CORRECTION FACTORS

When the **Include rarefaction effects** check box is selected in the physics interface **Particle Release and Propagation** section, a correction factor is applied to the drag force to account for the effects of a nonzero particle Knudsen number:

$$C_D = \frac{C_{D,C}}{S}$$

Where  $C_D$  (dimensionless) is the drag coefficient,  $C_{D,C}$  (dimensionless) is the drag coefficient under conditions of continuum flow (the limit as the particle Knudsen number  $\text{Kn}$  approaches zero), and  $S$  (dimensionless) is the correction factor that accounts for non-continuum flow.

For the following definitions of the drag force correction factor, the particle Knudsen number is defined as:

$$\text{Kn} = \frac{\lambda}{d_p}$$

where  $\lambda$  is the mean free path of gas molecules:

$$\lambda = \sqrt{\frac{\pi}{2p\rho}}\mu$$

where

- $p$  is the gas pressure (SI unit: Pa),
- $\rho$  is the gas density (SI unit:  $\text{kg}/\text{m}^3$ ),
- $d_p$  is the particle diameter (SI unit: m), and
- $\mu$  is the gas viscosity (SI unit: Pa s).

It is also possible to multiply this definition of the mean free path by a user-defined coefficient.



Use extra caution when interpreting expressions in terms of the Knudsen number from the literature, since other characteristic lengths may be used instead of the particle diameter; for example, the particle radius is sometimes used.

If the **Basset** model is used, the drag force correction factor is defined as:

$$S = \frac{1}{1 - (2 - \sigma_R)\frac{\text{Kn}}{\sigma_R}}$$

where  $\sigma_R$  (dimensionless) is the accommodation coefficient, the fraction of gas molecules that are assumed to reflect diffusely from the particle surface. The **Basset** model is appropriate for near-continuum flows.

If the **Epstein** model is used, the correction factor is defined as:

$$S = \frac{18\text{Kn}}{8 + \pi\sigma_R}$$

The **Epstein** model is appropriate for free molecular flows.

If the **Phillips** model is used, the correction factor is defined as:

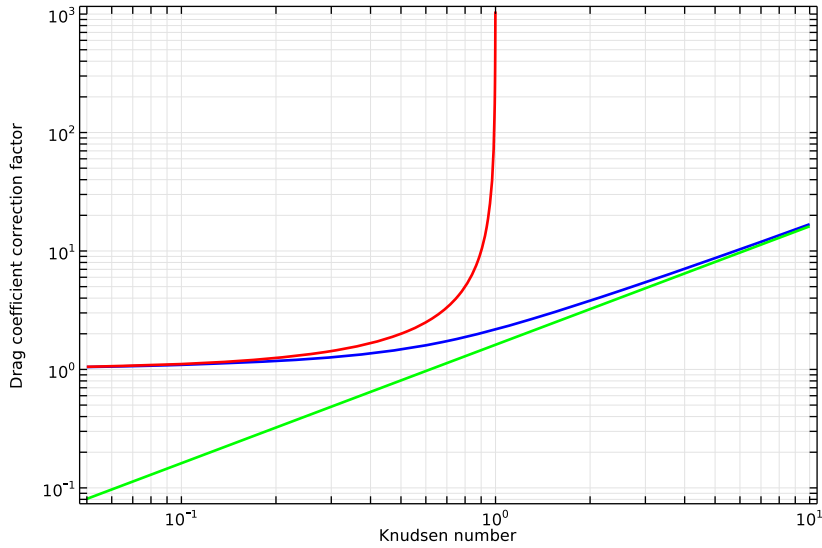
$$S = \frac{15 + 12c_1\text{Kn} + 9(c_1^2 + 1)\text{Kn}^2 + 18c_2(c_1^2 + 2)\text{Kn}^3}{15 - 3c_1\text{Kn} + c_2(8 + \pi\sigma_R)(c_1^2 + 2)\text{Kn}^2}$$

where  $c_1$  and  $c_2$  (dimensionless) are functions of the accommodation coefficient:

$$c_1 = \frac{2 - \sigma_R}{\sigma_R}$$

$$c_2 = \frac{1}{2 - \sigma_R}$$

The **Phillips** model has the same asymptotic behavior as the **Basset** and **Epstein** models, and is also usable at intermediate values of the particle Knudsen number. This asymptotic behavior is shown in [Figure 5-2](#).



*Figure 5-2: Comparison of the Basset, Epstein, and Phillips models for drag coefficient corrections in rarefied gases.*

The **Cunningham-Millikan-Davies** model uses three dimensionless, user-defined coefficients to define the correction factor:

$$S = 1 + \text{Kn} \left( C_1 + C_2 \exp\left(-\frac{C_3}{\text{Kn}}\right) \right)$$

## WALL CORRECTIONS

When the **Include wall corrections** check box is selected, the drag force is given by [Ref. 5](#) as

$$\mathbf{F}_{WC} = M\mathbf{F}_D$$
$$M = \left( 1 + \frac{1}{1 - \frac{9}{16}\alpha + \frac{1}{8}\alpha^3 - \frac{45}{256}\alpha^4 - \frac{1}{16}\alpha^5} \right) (I - P(\mathbf{n})) + \frac{1}{1 - \frac{9}{8}\alpha + \frac{1}{2}\alpha^3} P(\mathbf{n}) \quad (5-1)$$
$$\alpha = \frac{r_p}{l_W}$$

where

- $I$  is the identity matrix
- $P(\mathbf{n})$  is the projection operator onto the wall normal,  $\mathbf{n}$
- $r_p$  is the radius of the particle (SI unit: m)
- $l_W$  is the distance from the center of the particle to the nearest wall (SI unit: m)

The first term in [Equation 5-1](#) applies to the component of the relative particle velocity parallel to the nearest wall; the second term applies to the component of the relative velocity normal to this wall. The correction factors become more prominent as the wall distance becomes comparable to the particle radius.

## GRAVITY FORCE

The [Gravity Force](#) is given by:

$$\mathbf{F}_g = m_p \mathbf{g} \frac{(\rho_p - \rho)}{\rho_p}$$

where  $\rho$  is the density of the surrounding fluid (SI unit:  $\text{kg}/\text{m}^3$ ) and  $\mathbf{g}$  (SI unit:  $\text{m}/\text{s}^2$ ) is the gravity vector.

### *Particle Motion in a Turbulent Flow*

---

When particles move in a high Reynolds number flow, they may interact with a large number of eddies that perturb their trajectories. The fluid velocity field used to perturb the particle trajectories is generally computed using one of the following numerical methods ([Ref. 6](#)): Reynolds-averaged Navier Stokes (RANS), direct numerical simulation (DNS), or large eddy simulation (LES).

Although DNS (and, to some degree, LES) are capable of resolving the fluid velocity field on sufficiently small length scales so that the particle trajectories can be computed deterministically, the relatively high computational cost of these numerical methods often makes them impractical for simulation of high Reynolds number flows, particularly in complicated geometries. Instead, in the following analysis, it will be assumed that the fluid velocity field has been computed using the RANS model, in which the mean flow is simulated and variations in fluid velocity are described via the turbulent kinetic energy  $k$  (SI unit:  $\text{m}^2/\text{s}^2$ ). When solving for the particle trajectories, the drag force is computed by first adding a term to the mean flow field based on the local value of the turbulent kinetic energy.

The built-in turbulent dispersion models are dependent on the turbulent kinetic energy and on the turbulent dissipation rate  $\epsilon$  (SI unit:  $\text{m}^2/\text{s}^3$ ). Therefore, the accompanying fluid flow interface should use one of the following turbulence models, for which these variables are defined:

- k- $\epsilon$
- k- $\omega$
- SST
- Low Reynolds number k- $\epsilon$



For more information on the turbulence models see [Theory for the Turbulent Flow Interfaces](#) in the *CFD Module User's Guide*.

---

The following sections describe the available turbulent dispersion models. Each of these models can be used by selecting the appropriate option from the **Turbulent Dispersion** section of the settings window for the [Drag Force](#) node.

#### CONTINUOUS RANDOM WALK MODEL

In the **Continuous random walk** model ([Ref. 6,9,10](#)), the evaluation of the position and velocity of a fluid element chosen at random is a Markov process. For a time step  $dt$  (SI unit: s) taken by the Time-Dependent Solver, the general form for the change in the  $i$ th fluid velocity component  $du_i$  is

$$du_i = a_i(\mathbf{x}, \mathbf{u}, t)dt + b_{ij}(\mathbf{x}, \mathbf{u}, t)d\xi_j$$

where  $a_i$  and  $b_{ij}$  are coefficients yet to be specified,  $\mathbf{x}$  and  $\mathbf{u}$  are the position and velocity of the fluid element, and  $d\xi_j$  are the increments of a vector-valued Wiener process with independent components, which are uncorrelated Gaussian random

numbers with zero mean and variance  $dt$  (Ref. 9). A more in-depth discussion of the relevance of Weiner processes in turbulent dispersion is given in Ref. 11.

In homogeneous turbulence, the velocity perturbations  $du_i$  are computed by solving a classical Langevin equation (Ref. 9,10):

$$d\mathbf{u}_f = -\frac{\mathbf{u}_f}{\tau_L}dt + \sigma\sqrt{\frac{2}{\tau_L}}d\xi$$

where  $s$  (SI unit: m/s) is the fluctuating rms of the velocity perturbation in any direction. For isotropic turbulence computed using the k-ε model,

$$\sigma = \sqrt{u_{f,1}^2} = \sqrt{u_{f,2}^2} = \sqrt{u_{f,3}^2} = \sqrt{\frac{2k}{3}}$$

In general, the Lagrangian integral time scale  $\tau_L$  (SI unit: s) is given by (Ref. 6)

$$\tau_{L,i} = \int_0^\infty \frac{u_{f,i}(t)u_{f,i}(t+s)}{u_{f,i}(t)^2} ds$$

Essentially, the Lagrangian time scale characterizes the time interval size over which successively sampled random numbers should become uncorrelated. Typically  $\tau_L$  is estimated using the functional form

$$\tau_L = C_L \frac{k}{\varepsilon}$$

where the dimensionless coefficient  $C_L$  is on the order of unity. Typical values of  $C_L$  have been reported in the interval [0.2,0.96] (Ref. 6) and as low as 1/7 (Ref. 10).

It is convenient to express the Langevin equation in terms of a dimensionless velocity perturbation of the form

$$\mathbf{u}'_f = \frac{\mathbf{u}_f}{\sigma}$$

The components of  $\mathbf{u}'_f$  are stored for each model particle as auxiliary dependent variables, requiring one additional degree of freedom per particle per space dimension (including out-of-plane DOFs). The normalized Langevin equation then takes the form

$$d\mathbf{u}'_f = -\frac{\mathbf{u}'_f}{\tau_L}dt + \sqrt{\frac{2}{\tau_L}}d\xi$$



A drift correction term is added to the normalized Langevin equation to reduce non-physical diffusion of particles:

$$d\mathbf{u}'_f = -\frac{\mathbf{u}'_f}{\tau_L}dt + \sqrt{\frac{2}{\tau_L}}d\xi + \delta\mathbf{u}'_f$$

For isotropic turbulence, the following expression for the velocity perturbation, including the drift term, is given (Ref. 10):

$$d\mathbf{u}'_f = -\frac{\mathbf{u}'_f}{\tau_L}dt + \sqrt{\frac{2}{\tau_L}}d\xi + \frac{1}{3\sigma\partial\mathbf{x}1} \frac{dt}{St}$$

The particle Stokes number  $St$  (dimensionless) is defined as

$$St = \frac{\tau_p}{\tau_L}$$

where the particle relaxation time scale  $\tau_p$  (SI unit: s) or particle velocity response time depends on the drag law being used (see the [Drag Force](#) section) and on rarefaction effects applied to the drag force, if any (see the [Drag Force Correction Factors](#) section).

#### *Anisotropic Turbulence*

If the **Continuous random walk** model is used and the **Include anisotropic turbulence in boundary layers** check box is selected, the velocity perturbations are computed using different expressions in the streamwise, spanwise, and wall normal directions at the position of each particle.

In the following, the subscripts 1, 2, and 3 refer to the streamwise, spanwise, and wall normal directions, respectively. The streamwise direction is parallel to the fluid velocity. The wall normal direction points away from the nearest surface with a [Wall](#) boundary condition. Boundaries with the [Inlet](#), [Outlet](#), and [Symmetry](#) boundary conditions are not considered walls for the purpose of computing the wall normal direction. The spanwise direction is orthogonal to the wall normal and streamwise directions. Thus the streamwise, spanwise, and wall normal directions form an orthonormal basis,

$$\begin{aligned}\mathbf{n}_3 &= \mathbf{n}_1 \times \mathbf{n}_2 \\ \mathbf{n}_2 &= \mathbf{n}_3 \times \mathbf{n}_1 \\ \mathbf{n}_1 &= \mathbf{n}_2 \times \mathbf{n}_3\end{aligned}$$

The anisotropic turbulence model differs from the isotropic **Continuous random walk** model when the particles are sufficiently close to walls such that  $y^+ < 100$ , where  $y^+$  is the wall distance in dimensionless units,

$$y^+ = x_3 \frac{u^*}{\rho \mu}$$

$x_3$  (SI unit: m) is the distance to the nearest wall,  $\rho$  (SI unit: kg/m<sup>3</sup>) is the fluid density,  $\mu$  (SI unit: Pa s) is the fluid dynamic viscosity, and  $u^*$  (SI unit: m/s) is the friction velocity at the nearest wall. Usually  $u^*$  is computed by one of the turbulent flow physics interfaces.

The normalized Langevin equations in the streamwise, spanwise, and normal directions, respectively, are

$$\begin{aligned} d\left(\frac{u_{f1}}{u_{f1, \text{rms}}}\right) &= -\frac{u_{f1}}{u_{f1, \text{rms}} \tau_L} dt + \sqrt{\frac{2}{\tau_L}} d\xi_1 \\ d\left(\frac{u_{f2}}{u_{f2, \text{rms}}}\right) &= -\frac{u_{f2}}{u_{f2, \text{rms}} \tau_L} dt + \sqrt{\frac{2}{\tau_L}} d\xi_2 \\ d\left(\frac{u_{f3}}{u_{f3, \text{rms}}}\right) &= -\frac{u_{f3}}{u_{f3, \text{rms}} \tau_L} dt + \sqrt{\frac{2}{\tau_L}} d\xi_3 + \frac{du_{f3, \text{rms}}}{dx_3} \frac{1}{1 + \text{St}} dt \end{aligned}$$

where the rms velocity perturbations (Ref. 10) are

$$\begin{aligned} \frac{u_{f1, \text{rms}}}{u^*} &= \frac{0.40y^+}{1.0 + 0.0239(y^+)^{1.496}} \\ \frac{u_{f2, \text{rms}}}{u^*} &= \frac{0.19y^+}{1.0 + 0.0361(y^+)^{1.496}} \\ \frac{u_{f3, \text{rms}}}{u^*} &= \frac{0.0116(y^+)^2}{1.0 + 0.23y^+ + 0.00140(y^+)^{2.421}} \end{aligned}$$

The Lagrangian time scale  $\tau_L$  (SI unit: s) is a piecewise function of the wall distance,

$$\tau_L = \begin{cases} \frac{10\rho\mu}{u^*} & y^+ \leq 5 \\ \frac{10\rho\mu}{(u^*)^2} (7.122 + 0.5731y^+ - 0.00129(y^+)^2) & y^+ > 5 \end{cases}$$

The dimensionless velocity perturbations are plotted in the anisotropic region in Figure 5-3. The Lagrangian time scale is plotted in Figure 5-4.

In all other respects, the effect of these velocity perturbation terms on the particle trajectories is the same as for the isotropic **Continuous random walk** model.

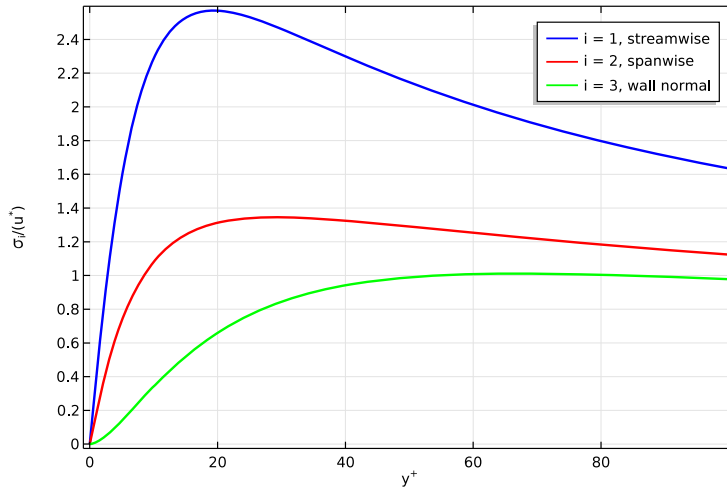


Figure 5-3: Dimensionless rms velocity perturbations in the streamwise, spanwise, and wall normal directions for modeling anisotropic turbulence.

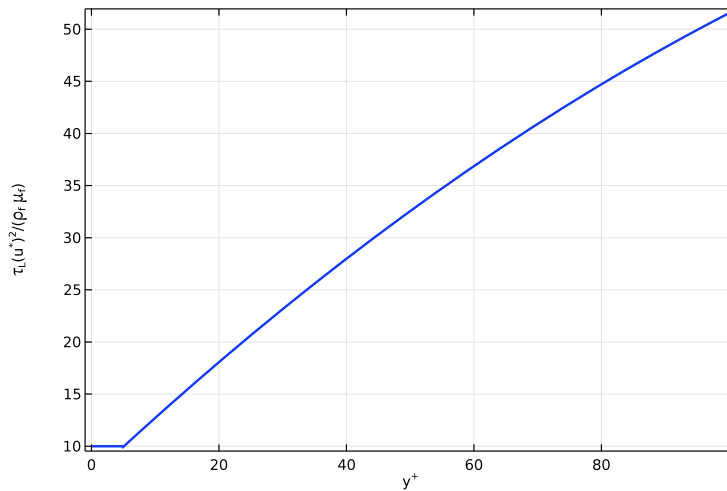


Figure 5-4: Lagrangian time scale as a function of wall distance for modeling anisotropic turbulence.

## DISCRETE RANDOM WALK MODEL

In the **Discrete random walk** model (Ref. 6,12,13), the components of the velocity perturbation are not computed by solving a Langevin equation, but are instead given instantaneous values

$$u_{f,i} = \zeta_i \sqrt{\frac{2k}{3}}$$

where  $\zeta_i$  (dimensionless) are uncorrelated Gaussian random numbers with unit variance (Ref. 13).

If **Discrete random walk** is selected from the **Turbulent dispersion model** list, then the  $\zeta_i$  are recomputed at every time step taken by the solver, whenever particles are released, and whenever particles interact with boundaries. If the time step taken by the solver is made arbitrarily small, the amount of turbulent dispersion in the model will decrease to non-physically small values.

To avoid such non-physical behavior, when using the **Discrete random walk** option it is recommended to specify a time step size approximately equal to the average interaction time between a particle and an eddy in the flow. One estimate of this interaction time is the eddy lifetime  $\tau_e$  (SI unit: s), given by (Ref. 6)

$$\tau_e = 2\tau_L$$

where  $\tau_L$  is the Lagrangian response time. If the option **Discrete random walk, variable time step** is selected, then the  $\zeta_i$  are only recomputed whenever a particle has existed for an integer number of eddy lifetimes. This is accomplished by allocating a dimensionless auxiliary dependent variable  $N$  with initial value 0. The value of  $N$  is integrated over time using the equation

$$\frac{dN}{dt} = \frac{1}{\tau_e}$$

Then `floor(N)` is used instead of the instantaneous time  $t$  as an argument in the random number seed used to compute the Gaussian random numbers  $\zeta_i$ . The resulting distribution of particle velocity values will then statistically converge and have nonzero variance, even if the time step taken by the solver is arbitrarily small.

### *Particle Motion in a Shear Flow*

---

Particles in a non-uniform velocity field are subjected to lift and drag forces. The drag force acts in the direction opposite the relative velocity of the particle with respect to

the fluid. The lift force acts along the direction of the gradient of the fluid velocity. These forces change significantly as the particle approaches a wall.

Use the **Lift Force** feature to exert a lift force on the particles. Two formulations of the lift force are available. Both expressions for the lift force are valid when the angular velocity of the particle can be neglected and a no-slip boundary condition applies at the particle surface.

The following sections describe the available lift force models. Each of these models can be used by selecting the appropriate option from the **Lift law** list in the settings window for the **Lift Force** node. Use the **Saffman** lift force when particles are in a non-uniform velocity field far from walls. The **Wall induced** lift force applies to particles close to the walls of a flow channel.

### SAFFMAN LIFT FORCE

The **Saffman** lift force  $\mathbf{F}_L$  (SI unit: N), described in [Ref. 7](#), is given by

$$\mathbf{F}_L = -81,2r_p^2\mathbf{L}_v\sqrt{\mu\rho\frac{|\mathbf{u}-\mathbf{v}|}{|\mathbf{L}_v|}}$$

$$\mathbf{L}_v = (\mathbf{u}-\mathbf{v})\times[\nabla\times(\mathbf{u}-\mathbf{v})]$$

where

- $r_p$  is the particle radius (SI unit: m)
- $\mu$  is fluid viscosity (SI unit: Pa s)
- $\mathbf{v}$  is the velocity of the particle (SI unit: m/s)
- $\mathbf{u}$  is the fluid velocity (SI unit: m/s)

The **Saffman** lift force is only nonzero for particles that have not yet been entrained in the surrounding fluid. Once a particle has reached equilibrium with the surrounding fluid velocity, the Saffman lift force is zero. To model the inertial lift force on particles in a flow channel, which is typically nonzero even after the particles become entrained in the flow, consider using the **Wall induced** lift force described in the following section.

### WALL INDUCED LIFT FORCE

The **Wall induced** lift force model described in [Ref. 8](#) uses a first order correction to the velocity profile at the channel walls and a second order correction at the surface of the particle in order to account for higher-order derivatives of the fluid velocity, compared to the **Saffman** model. It is given by

$$\mathbf{F}_L = \rho \frac{r_p^4}{D^2} \beta (\beta G_1(s) + \alpha G_2(s)) \mathbf{n}$$

$$\beta = |D(\mathbf{n} \cdot \nabla) \mathbf{u}_p|$$

$$\gamma = \left| \frac{D^2}{2} (\mathbf{n} \cdot \nabla)^2 \mathbf{u}_p \right|$$

$$\mathbf{u}_p = (I - (\mathbf{n} \otimes \mathbf{n})) \mathbf{u}$$

where

- $r_p$  is the particle radius (SI unit: m)
- $\mu$  is fluid viscosity (SI unit: Pa s)
- $\mathbf{u}$  is the fluid velocity (SI unit: m/s)
- $I$  is the identity matrix (dimensionless),
- $\mathbf{n}$  is the wall normal at the nearest point on the reference wall (dimensionless)



Here the term “reference wall” refers to the set of boundaries selected in the **Parallel Boundary 1** section in the settings window for the **Lift Force** node.

- $D$  is the distance between the channel walls (SI unit: m)
- $s$  is the nondimensionalized distance from the particle to the reference wall, divided by  $D$  so that  $0 < s < 1$  for particles in the channel
- $G_1$  and  $G_2$  are functions of nondimensionalized wall distance  $s$  as given in [Ref. 8](#) (dimensionless). They are plotted in [Figure 5-5](#) and [Figure 5-6](#) below.

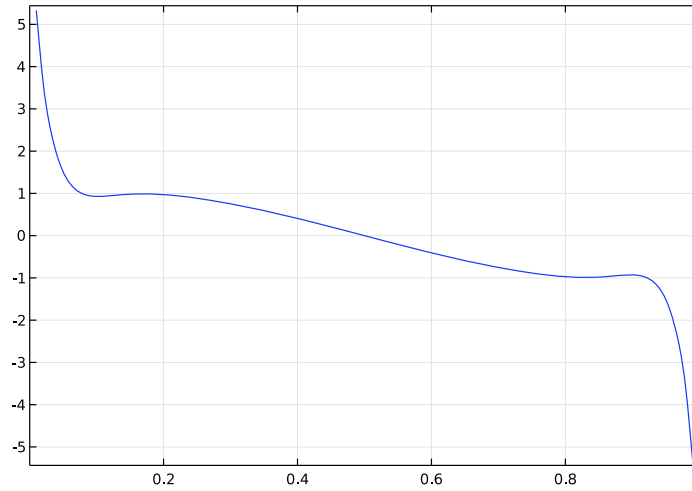


Figure 5-5:  $G_1$  coefficient for the lift force.

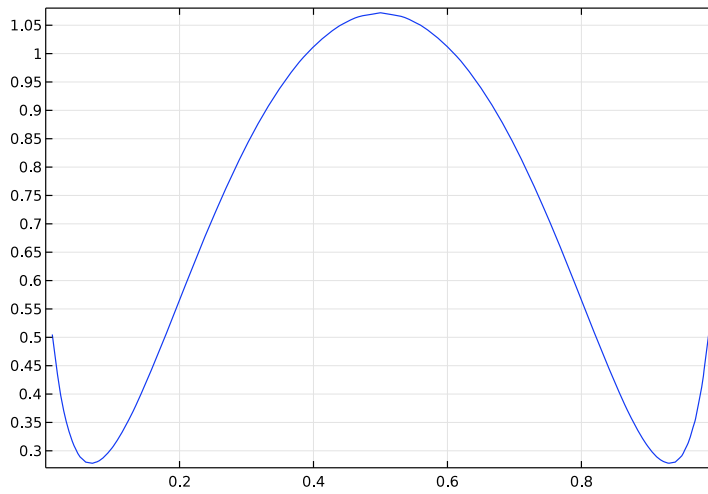


Figure 5-6:  $G_2$  coefficient for the lift force.

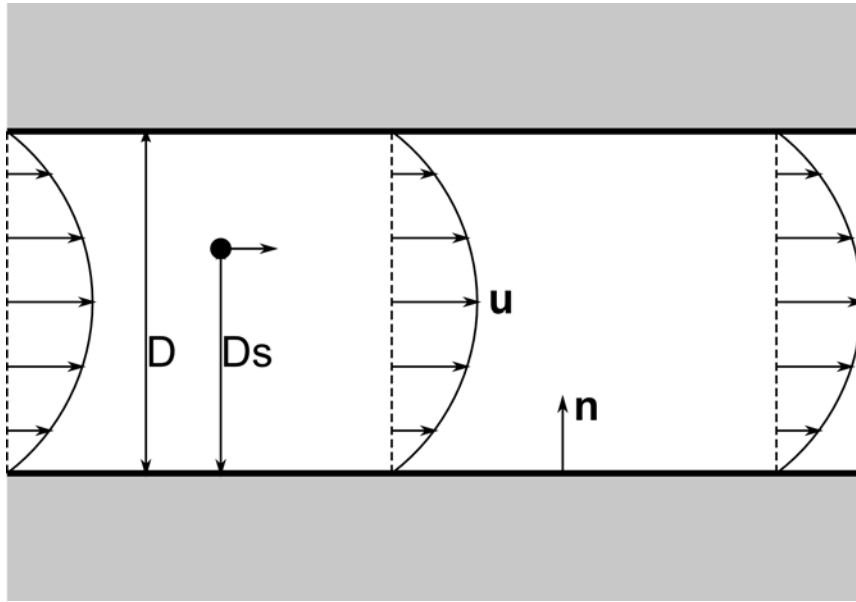


Figure 5-7: Dimensions used to define the wall induced lift force. Here the bottom wall has been selected in the Parallel Boundary 1 section and the top wall has been selected in the Parallel Boundary 2 section.

### Additional Forces

In addition to the [Drag Force](#) and [Gravity Force](#) described in [Particle Motion in a Fluid](#), the following predefined forces can be added to influence particle motion.

- [Brownian Force](#)
- [Electric Force](#)
- [Magnetic Force](#)
- [Dielectrophoretic Force](#)
- [Acoustophoretic Force](#)
- [Thermophoretic Force](#)
- [Magnetophoretic Force](#)

### BROWNIAN FORCE

The [Brownian Force](#) feature adds the following contribution to  $\mathbf{F}_{\text{ext}}$ :

$$\mathbf{F}_{\text{ext}} = \zeta \sqrt{\frac{12\pi k_B \mu T r_p}{\Delta t}} \quad (5-2)$$



where

- $\Delta t$  (SI unit: s) is the time step taken by the solver
- $r_p$  (SI unit: m) is the particle radius
- $T$  (SI unit: K) is the absolute fluid temperature
- $\mu$  (SI unit: Pa s) is the fluid dynamic viscosity,
- $k_B = 1.3806488 \times 10^{-23}$  J/K is the Boltzmann constant, and
- $\zeta$  is a normally distributed random number with a mean of zero and unit standard deviation.

As explained in [Ref. 16](#), independent values for  $\zeta$  are chosen in all directions. A different value of  $\zeta$  is created for each particle, at each time step for each component of the Brownian force. The Brownian force leads to spreading of particles from regions of high particle density to low density.



The fluid viscosity required by the Brownian force must be consistent with the one supplied in the Drag force. Furthermore, the specific form of the Brownian force given above is only consistent if the **Drag law** is set to **Stokes**.

## ELECTRIC FORCE

The [Electric Force](#) feature adds the following contribution to  $\mathbf{F}_{\text{ext}}$ :

$$\mathbf{F}_{\text{ext}} = eZ\mathbf{E}$$

where  $e = 1.602176565 \times 10^{-19}$  C is the elementary charge,  $Z$  is the charge number (dimensionless), and  $\mathbf{E}$  is the electric field (SI unit: V/m). The electric field can be given directly or computed from an electric potential:

$$\mathbf{E} = -\nabla V$$

When the electric field is computed in the frequency domain it is complex valued. The field must be cast into a real value that depends on the angular frequency and the simulation time:

$$\mathbf{E}(t) = \text{real}(\tilde{\mathbf{E}} \exp(j(\omega t + \phi_0)))$$

where  $\tilde{\mathbf{E}}$  is the complex-valued electric field,  $\omega$  is the angular frequency,  $\phi_0$  is the initial phase angle, and  $t$  is time.

## MAGNETIC FORCE

The [Magnetic Force](#) feature adds the following contribution to  $\mathbf{F}_{\text{ext}}$ :

$$\mathbf{F}_{\text{ext}} = eZ(\mathbf{v} \times \mathbf{B})$$

When the magnetic flux density is computed in the frequency domain it is complex valued. The field must be cast into a real value that depends on the angular frequency and the simulation time:

$$\mathbf{B}(t) = \text{real}(\tilde{\mathbf{B}} \exp(j(\omega t + \phi_0)))$$

where  $\tilde{\mathbf{B}}$  is the complex-valued electric field,  $\omega$  is the angular frequency,  $\phi_0$  is the initial phase angle, and  $t$  is time.

## DIELECTROPHORETIC FORCE

The [Dielectrophoretic Force](#) feature adds the following contribution to  $\mathbf{F}_{\text{ext}}$  for the case of a static electric field:

$$\mathbf{F}_{\text{ext}} = 2\pi r_p^3 \epsilon_0 \text{real}(\epsilon^*) \text{real}\left(\frac{\epsilon_p^* - \epsilon^*}{\epsilon_p^* + 2\epsilon^*}\right) \nabla |\mathbf{E}|^2$$

where  $\epsilon^*$  (dimensionless) is the complex relative permittivity of the fluid,  $\epsilon_p^*$  (dimensionless) is the complex relative permittivity of the particle, and  $\mathbf{E}$  (SI unit: V/m) is the electric field. When computing the dielectrophoretic force exerted by a stationary electric field, the complex relative permittivity becomes the real-valued relative permittivity. In the case that the electric field is computed in the frequency domain, the following is added:

$$\mathbf{F}_{\text{ext}} = 2\pi r_p^3 \epsilon_0 \text{real}(\epsilon^*) \text{real}\left(\frac{\epsilon_p^* - \epsilon^*}{\epsilon_p^* + 2\epsilon^*}\right) \nabla |\mathbf{E}_{\text{rms}}|^2$$

where  $\mathbf{E}_{\text{rms}}$  denotes the root mean square electric field. For fields that are computed in the frequency domain, the complex permittivity can be expressed as

$$\epsilon^* = \epsilon - \frac{i\sigma}{\omega}$$

where  $\epsilon$  (SI unit: F/m) is the permittivity,  $\sigma$  (SI unit: S/m) is the electrical conductivity, and  $\omega$  (SI unit: rad/s) is the angular frequency of the electric field.

The **Shell** feature can be added to the **Dielectrophoretic Force** node to model the dielectrophoretic force on particles with thin shells. The complex permittivity of the shell can differ from the permittivity of the rest of the particle. When computing the dielectrophoretic force, the complex permittivity of the particle is replaced by the equivalent complex relative permittivity  $\epsilon_{\text{eq}}^*$  of a homogeneous particle comprising both the shell and the interior of the particle:

$$\epsilon_{\text{eq}}^* = \epsilon_s^* \frac{\left(\frac{r_0}{r_i}\right)^3 + 2\left(\frac{\epsilon_p^* - \epsilon_s^*}{\epsilon_p^* + 2\epsilon_s^*}\right)}{\left(\frac{r_0}{r_i}\right)^3 - \left(\frac{\epsilon_p^* - \epsilon_s^*}{\epsilon_p^* + 2\epsilon_s^*}\right)}$$

where  $r_0$  and  $r_i$  (SI unit: m) are the outer and inner radii of the shell, respectively;  $\epsilon_p^*$  (dimensionless) is the complex relative permittivity of the particle, and  $\epsilon_s^*$  (dimensionless) is the complex relative permittivity of the outer shell. If multiple shells are added, the shells are applied in the order in which they appear in the Model Builder, with the first shell being the innermost. The present treatment of spherical particles with thin shells is described in [Ref. 3](#).

#### ACOUSTOPHORETIC FORCE

The **Acoustophoretic Force** is a special subclass of the CFD forces, which are the acoustic radiation forces on small particles. The present theory is derived for inviscid fluids and small particles. This implies that the expressions are valid for particles of diameter  $d_p = 2r_p$  much smaller than the acoustic wavelength  $\lambda$  and of size larger than the acoustic boundary layer thickness  $\lambda_{\text{bl}}$ . The last condition is not necessary for buoyant particles (of same density  $\rho_p$  as the fluid density  $\rho_0$ ). The present expressions are based on the work of Gorkov ([Ref. 14](#)) which is reviewed by Bruus (see [Ref. 15](#)).

Following the references the radiation force on a particle is given as

$$\mathbf{F}_{\text{aco}} = -\nabla U^{\text{rad}}$$

where  $U^{\text{rad}}$  is the radiation potential given by

$$U^{\text{rad}} = V_p \left[ f_1 \frac{1}{2\rho_0 c_0^2} \langle p_{\text{in}}^2 \rangle - f_2 \frac{3}{4} \rho_0 \langle v_{\text{in}}^2 \rangle \right]$$

$$f_1 = 1 - \frac{\rho_0 c_0^2}{\rho_p c_p^2} = 1 - \frac{\beta_p}{\beta_0} = 1 - \frac{K_0}{K_p}$$

$$f_2 = \frac{2(\rho_p - \rho_0)}{2\rho_p + \rho_0}$$

where

- $p_{\text{in}}$  (SI unit: Pa) is the incident (external) pressure field,
- $v_{\text{in}}$  (SI unit: m/s) is the incident particle velocity amplitude,
- $c_0$  (SI unit: m/s) is the speed of sound in the fluid,
- $c_p$  (SI unit: m/s) is the compressional speed of sound in the particle, and
- $\beta_p$  is the compressibility of the particle (reciprocal of bulk modulus =  $(K_p)^{-1}$ ).

The  $\langle \rangle$  terms are time averaged over one period.

In the case of an acoustic field derived from a time-harmonic analysis (frequency-domain analysis) where the dependent variables are complex valued, the time-averaged expressions simplify as

$$\langle p_{\text{in}}^2 \rangle = \frac{1}{2} \langle p_{\text{in}} p_{\text{in}}^* \rangle$$

$$\langle v_{\text{in}}^2 \rangle = \frac{1}{2} \langle v_{\text{in}} v_{\text{in}}^* \rangle = \frac{1}{2} \langle \mathbf{v}_{\text{in}} \cdot \mathbf{v}_{\text{in}}^* \rangle$$

where  $*$  is the complex conjugate. For the case of pressure acoustics the velocity is simply given as

$$\mathbf{v}_{\text{in}} = -\frac{1}{i\omega\rho_0} \nabla p$$

Thus the present acoustic radiation forces are only space dependent and not time-dependent as they are time averaged.

### THERMOPHORETIC FORCE

The [Thermophoretic Force](#) accounts for a force on a particle due to gradients in the temperature of the background fluid. The thermophoretic force is defined as:

$$\mathbf{F}_{tp} = -\frac{6\pi d_p \mu^2 C_s \Lambda \nabla T}{\rho(2\Lambda + 1)T}$$

$$\Lambda = \frac{k}{k_p}$$

where

- $k$  (SI unit: W/(m K)) is the thermal conductivity of the fluid,
- $k_p$  (SI unit: W/(m K)) is the particle thermal conductivity,
- $T$  (SI unit: K) is the fluid temperature,
- $d_p$  (SI unit: m) is the particle diameter,
- $\rho$  (SI unit: kg/m<sup>3</sup>) is the fluid density, and
- $C_s$  is a dimensionless constant equal to 1.17.

This expression for the thermophoretic force is valid for continuum flows, in which the particle Knudsen number is very small. The thermophoretic force makes particles move from hotter to cooler regions. This is why dust tends to settle in a corner of the kitchen, furthest away from the oven.

#### RAREFACTION EFFECTS IN THERMOPHORESIS

If the **Include rarefaction effects** check box is selected in the physics interface **Particle Release and Propagation** section, other expressions for the thermophoretic force are available. The **Epstein** model is equivalent to the expression for continuum flow.

If the **Waldmann** model is selected, the thermophoretic force is defined as:

$$\mathbf{F}_{tp} = -\frac{8}{15} \sqrt{\frac{2\pi M_g}{k_B T}} k r_p^2 \nabla T$$

where

- $M_g$  (SI unit: kg/mol) is the gas molar mass,
- $k_B = 1.3806488 \times 10^{-23}$  J/K is the Boltzmann constant, and
- $r_p$  (SI unit: m) is the particle radius.

The **Waldmann** model is appropriate for free molecular flows.

If the Talbot model is selected, the thermophoretic force is defined as:

$$\mathbf{F}_{tp} = -\frac{6\pi d_p \mu^2}{\rho} \frac{C_s(\Lambda + C_t \text{Kn})}{(1 + 3C_m \text{Kn})(1 - 2\Lambda + 2C_t \text{Kn})} \frac{\nabla T}{T}$$

If the Linearized BGK model is selected, the thermophoretic force is expressed using the analysis of the linearized Bhatnagar-Gross-Krook (BGK) and S model kinetic equations by Beresnev and Chernyak (Ref. 17). The thermophoretic force is defined in terms of the free-molecular limit,

$$\mathbf{F}_{tp} = \frac{\left[ \alpha_E + \frac{15}{8} \sqrt{\pi} (1 - \alpha_E) \frac{\Lambda}{R} \right] \phi_1 + \alpha_E \Lambda \phi_2}{\left[ \alpha_E + \frac{15}{8} \sqrt{\pi} (1 - \alpha_E) \frac{\Lambda}{R} \right] \phi_3 + \left[ \alpha_E + \frac{5}{8} \sqrt{\pi} (3 - \alpha_E) \frac{\Lambda}{R} \right] \phi_4} \mathbf{F}_W$$

where  $\mathbf{F}_W$  (SI unit: N) is the thermophoretic force as calculated in the Waldmann model,  $R$  is a dimensionless number defined as

$$R = \frac{\sqrt{\pi}}{3\text{PrKn}}$$

where Pr (dimensionless) is the fluid Prandtl number,

$$\text{Pr} = \frac{\mu C_p}{k}$$

and  $\alpha_E$  (dimensionless) is the energy accommodation coefficient, with  $\alpha_E = 1$  and  $\alpha_E = 0$  corresponding to diffuse and specular reflection of all gas molecules at the surface, respectively. The dimensionless coefficients  $\phi_i$  are defined as

$$\phi_i = f_{i1} + (1 - \alpha_T) f_{i2}$$

where the dimensionless terms  $f_{ij}$  are defined as interpolation functions of  $R$  (Ref. 17).

<b>R</b>	<b>f<sub>11</sub></b>	<b>f<sub>12</sub></b>	<b>f<sub>21</sub></b>	<b>f<sub>22</sub></b>
1000	$-1.324 \times 10^{-5}$	$1.317 \times 10^{-5}$	$1.366 \times 10^{-9}$	$4.924 \times 10^{-8}$
800	$-2.067 \times 10^{-5}$	$2.055 \times 10^{-5}$	$2.677 \times 10^{-9}$	$9.601 \times 10^{-8}$
600	$-3.673 \times 10^{-5}$	$3.645 \times 10^{-5}$	$6.379 \times 10^{-9}$	$2.270 \times 10^{-7}$
400	$-8.256 \times 10^{-5}$	$8.159 \times 10^{-5}$	$2.175 \times 10^{-8}$	$7.619 \times 10^{-7}$
200	$-3.291 \times 10^{-4}$	$3.215 \times 10^{-4}$	$1.791 \times 10^{-7}$	$5.997 \times 10^{-6}$
100	$-1.307 \times 10^{-3}$	$1.248 \times 10^{-3}$	$1.502 \times 10^{-6}$	$4.643 \times 10^{-5}$
80	$-2.036 \times 10^{-3}$	$1.921 \times 10^{-3}$	$2.987 \times 10^{-6}$	$8.917 \times 10^{-5}$

<b>R</b>	<b>f<sub>11</sub></b>	<b>f<sub>12</sub></b>	<b>f<sub>21</sub></b>	<b>f<sub>22</sub></b>
60	$-3.598 \times 10^{-3}$	$3.331 \times 10^{-3}$	$7.225 \times 10^{-6}$	$2.055 \times 10^{-4}$
40	$-8.004 \times 10^{-3}$	$7.139 \times 10^{-3}$	$2.439 \times 10^{-5}$	$6.551 \times 10^{-4}$
20	$-3.115 \times 10^{-2}$	$2.505 \times 10^{-2}$	$1.066 \times 10^{-4}$	$4.417 \times 10^{-3}$
10	-0.1109	$7.525 \times 10^{-2}$	$-1.852 \times 10^{-3}$	$2.639 \times 10^{-2}$
8	-0.1636	0.1039	$-6.174 \times 10^{-3}$	$4.513 \times 10^{-2}$
6	-0.2626	0.1524	$-2.244 \times 10^{-2}$	$8.749 \times 10^{-2}$
4	-0.4780	0.2443	-0.1013	0.2089
2	-1.058	0.4507	-0.7320	0.7674
1	-1.749	0.6623	-3.077	2.247
0.8	-1.952	0.7205	-4.501	3.051
0.6	-2.181	0.7851	-7.034	4.424
0.4	-2.439	0.8559	$-1.237 \times 10^1$	7.214
0.2	-2.717	0.9310	$-2.889 \times 10^1$	$1.562 \times 10^1$
0.1	-2.857	0.9678	$-6.220 \times 10^1$	$3.238 \times 10^1$
0.08	-2.884	0.9748	$-7.885 \times 10^1$	$4.073 \times 10^1$
0.06	-2.910	0.9816	$-1.066 \times 10^2$	$5.463 \times 10^1$
0.04	-2.835	0.9881	$-1.620 \times 10^2$	$8.239 \times 10^1$

<b>R</b>	<b>f<sub>31</sub></b>	<b>f<sub>32</sub></b>	<b>f<sub>41</sub></b>	<b>f<sub>42</sub></b>
1000	1	-0.9947	-1.004	0.9991
800	1	-0.9934	-1.006	0.9989
600	1	-0.9912	-1.007	0.9985
400	1	-0.9868	-1.011	0.9978
200	1	-0.9739	-1.022	0.9955
100	1	-0.9489	-1.044	0.9909
80	1	-0.9367	-1.055	0.9885
60	1	-0.9169	-1.074	0.9846
40	1	-0.8794	-1.110	0.9765
20	1	-0.7812	-1.218	0.9512
10	1	-0.6360	-1.415	0.8994
8	1	-0.5819	-1.504	0.8737
6	1	-0.5101	-1.635	0.8319
4	1	-0.4098	-1.847	0.7535

<b>R</b>	<b>f<sub>31</sub></b>	<b>f<sub>32</sub></b>	<b>f<sub>41</sub></b>	<b>f<sub>42</sub></b>
2	1	-0.2570	-2.226	0.5676
1	1	-0.1421	-2.533	0.357
0.8	1	-0.1142	-2.609	0.2962
0.6	1	-8.453 × 10 <sup>-2</sup>	-2.691	0.2265
0.4	1	-5.344 × 10 <sup>-2</sup>	-2.781	0.1482
0.2	1	-2.237 × 10 <sup>-2</sup>	-2.881	6.437 × 10 <sup>-2</sup>
0.1	1	-8.413 × 10 <sup>-3</sup>	-2.937	2.470 × 10 <sup>-2</sup>
0.08	1	-5.993 × 10 <sup>-3</sup>	-2.945	1.767 × 10 <sup>-2</sup>
0.06	1	-3.793 × 10 <sup>-3</sup>	-2.961	1.123 × 10 <sup>-2</sup>
0.04	1	-1.900 × 10 <sup>-3</sup>	-2.973	5.648 × 10 <sup>-3</sup>

Note that this data is tabulated assuming that the fluid Prandtl number is 2/3.

### MAGNETOPHORETIC FORCE

The [Magnetophoretic Force](#) causes motion of permeable particles toward regions where the magnetic field is stronger. The magnetophoretic force is applicable for particles which are charge neutral and have a different relative permeability than the background fluid. The magnetophoretic force is defined as:

$$\mathbf{F}_m = 2\pi r_p^3 \mu_0 \mu_r K \nabla |\mathbf{H}|^2 \quad (5-3)$$

where  $\mathbf{H}$  is the magnetic field,  $\mu_r$  is the fluid relative permeability,  $\mu_{r,p}$  is the particle relative permeability, and  $K$  is defined as:

$$K = \frac{\mu_{r,p} - \mu_r}{\mu_{r,p} + 2\mu_r}$$

The force is only valid for nonconducting particles, so additional force contributions due to eddy currents induced in particles are not taken into account.

#### Numerical Differentiation of Magnetic Field Components

As shown in [Equation 5-3](#), the magnetophoretic force is a function of derivatives of the components of the magnetic field  $\mathbf{H}$ . Depending on the physics interface used to compute the magnetic field, its derivatives may not be available.

For example, in instances of the Magnetic Fields interface, the degrees of freedom correspond to components of the magnetic vector potential  $\mathbf{A}$  (SI unit: Wb/m). The magnetic field can be derived from the magnetic vector potential using the relations



$$\mathbf{B} = \nabla \times \mathbf{A}$$

$$\mathbf{B} = \mu_0(\mathbf{H} + \mathbf{M})$$

where  $\mathbf{B}$  (SI unit: T) is the magnetic flux density and  $\mathbf{M}$  (SI unit: A/m) is the magnetization vector. Thus, derivatives of the magnetic field components correspond to second derivatives of the degrees of freedom, the components of  $\mathbf{A}$ . However, the second derivative is not defined for the vector (curl) elements that are used to discretize the components of  $\mathbf{A}$  in 3D or the in-plane components of  $\mathbf{A}$  in 2D. In these situations, the derivatives of the magnetic field components will be evaluated as zero everywhere, resulting in zero magnetophoretic force.

Therefore, depending on the formulation used to compute the magnetic potential, it may be necessary to take additional steps to ensure that the derivatives of the magnetic field components are evaluated correctly. A typical approach involves the definition of an intermediate PDE interface to store the values of the magnetic field components using scalar basis functions.

### *Erosion Theory*

---

The [Erosion](#) feature calculates the rate of erosive wear or the total mass removed per unit area due to the impact of particles on a surface. It includes the following models:

#### **FINNIE**

Finnie ([Ref. 19](#)) defined the volume removed from a surface as:

$$V = \frac{cMU^2}{4p\left(1 + \frac{mr^2}{I}\right)} [(\cos \alpha)^2] \quad \tan \alpha > \frac{P}{2}$$

$$V = \frac{cMU^2}{4p\left(1 + \frac{mr^2}{I}\right)} \frac{2}{P} \left[ \sin(2\alpha) - 2\frac{(\sin \alpha)^2}{P} \right] \quad \tan \alpha \leq \frac{P}{2}$$

The parameters are defined as follows:

- $c$  (dimensionless) is the fraction of particles cutting in an idealized manner.
- $M$  (SI unit: kg) is the total mass of eroding particles.
- $U$  (m/s) is the magnitude of the incident particle velocity.
- $p$  (Pa) is the Vickers hardness of the material.

- $m$  (SI unit: kg) is the mass of an individual particle hitting the surface.
- $r$  (SI unit: m) is the average particle radius.
- $I$  (SI unit: kg-m<sup>2</sup>) is the moment of inertia of an individual particle about its center of mass. For an isotropic sphere,  $I = 2mr^2/5$ .
- $\alpha$  (rad) is the angle of incidence, with  $\alpha = 0$  tangent to the surface and  $\alpha = \pi/2$  normal to the surface.
- $P$  is a dimensionless parameter, defined as  $P = K/(1+mr^2/I)$ , where  $K$  (dimensionless) is the ratio of vertical and horizontal forces acting on the particle.

In the Finnie model, particles are assumed to remove mass from the surface via an idealized cutting mechanism. It does not predict any erosive wear by particles at normal incidence to a surface, and is recommended for modeling erosion of ductile materials by particles at small angles of incidence.

#### **E/CRC**

The E/CRC model defines the erosion rate in terms of the ratio of mass lost by the surface to mass of incident particles:

$$E = CF_s(\text{BH})^{-0.59} \left( \frac{v}{1[\text{m/s}]} \right)^n F(\alpha)$$

$$F(\alpha) = 5.40\alpha - 10.11\alpha^2 + 10.93\alpha^3 - 6.33\alpha^4 + 1.42\alpha^5$$

where  $C$  is an erosion model coefficient,  $F_s$  is the particle shape coefficient, and  $\text{BH}$  is the Brinell hardness of the wall material (all dimensionless). The angle of incidence  $\alpha$  is measured in radians.

#### **OKA**

The Oka model defines the volume of surface material removed per unit mass of incident particles (in units of mm<sup>3</sup>/kg) as:

$$E(\alpha) = g(\alpha)E_{90}$$

where  $E(\alpha)$  is the etch rate,  $E_{90}$  is the etch rate at normal incidence, and  $g(\alpha)$  is an angular dependence function:

$$g(\alpha) = (\sin \alpha)^{n_1} \left[ 1 + \frac{H_v}{1[\text{GPa}]} (1 - \sin \alpha) \right]^{n_2}$$

$$E_{90} = \left( \frac{Hv}{1[\text{GPa}]} \right)^{k_1} \left( \frac{v}{v'} \right)^{k_2} \left( \frac{D}{D'} \right)^{k_3}$$

where  $v'$  is the reference velocity;  $D'$  is the reference diameter; the dimensionless parameters  $n_1$ ,  $n_2$ , and  $k_2$  are defined as

$$n_1 = s_1 \left( \frac{Hv}{1[\text{GPa}]} \right)^{q_1}$$

$$n_2 = s_2 \left( \frac{Hv}{1[\text{GPa}]} \right)^{q_2}$$

$$k_2 = 2.3 \left( \frac{Hv}{1[\text{GPa}]} \right)^{0.038}$$

and the remaining dimensionless parameters  $s_1$ ,  $s_2$ ,  $q_1$ ,  $q_2$ ,  $K$ ,  $k_1$ , and  $k_3$  are constants.

#### **DNV**

The DNV model defines the erosion rate in terms of the ratio of mass lost by the surface to mass of incident particles:

$$E = K \left( \frac{v}{1[\text{m/s}]} \right)^{-n} F(\alpha)$$

$$F(\alpha) = 9.370\alpha - 42.295\alpha^2 + 110.864\alpha^3 - 175.804\alpha^4 + 170.137\alpha^5 - 98.398\alpha^6 + 31.211\alpha^7 - 4.170\alpha^8$$

where  $K$  and  $n$  are dimensionless constants that depend on the surface material.

#### *Droplet Breakup Theory*

---

It is possible to model the breakup of liquid droplets using the [Droplet Breakup](#) feature. This feature supports two subfeatures, [Kelvin-Helmholtz Breakup Model](#) (added by default) and [Rayleigh-Taylor Breakup Model](#), which correspond to different mechanisms for breakup. It is possible to include either breakup model separately or to use both models at once.

## DIMENSIONLESS NUMBERS

When analyzing the mechanisms of droplet breakup, several dimensionless parameters are useful. These parameters are summarized in [Table 5-1](#).

TABLE 5-1: DIMENSIONLESS QUANTITIES USED IN DROPLET BREAKUP MODELS

SYMBOL	NAME	EXPRESSION
$We_g$	Gas Weber number	$We_g = \frac{\rho U_{rel}^2 r_p}{\sigma_p}$
$We_l$	Liquid Weber number	$We_l = \frac{\rho_p U_{rel}^2 r_p}{\sigma_p}$
$Re_l$	Droplet Reynolds number	$Re_l = \frac{\rho_p r_p U_{rel}}{\mu_p}$
$Z$	Ohnesorge number	$Z = \frac{\sqrt{We_l}}{Re_l}$
$T$	Taylor number	$T = Z \sqrt{We_g}$

where

- $\rho$  (SI unit:  $\text{kg/m}^3$ ) is the gas density,
- $\rho_p$  (SI unit:  $\text{kg/m}^3$ ) is the droplet density,
- $U_{rel}$  (SI unit:  $\text{m/s}$ ) is the relative speed between the droplet and gas,
- $r_p$  (SI unit:  $\text{m}$ ) is the droplet radius,
- $\sigma_p$  (SI unit:  $\text{N/m}$ ) is the surface tension, and
- $\mu_p$  (SI unit:  $\text{Pa s}$ ) is the droplet dynamic viscosity.

## KELVIN-HELMHOLTZ BREAKUP

The [Kelvin-Helmholtz Breakup Model](#) is based on a first-order linear analysis of Kelvin-Helmholtz instability in a cylindrical jet of fluid. For a parent droplet of radius  $r_0$  (SI unit:  $\text{m}$ ), the wavelength  $\Lambda_{KH}$  (SI unit:  $\text{m}$ ) of the fastest-growing disturbance on the surface of the particle is

$$\Lambda_{KH} = \frac{9.02 r_0 (1 + 0.45 \sqrt{Z_l}) (1 + 0.4 T^{0.7})}{(1 + 0.865 We_g^{1.67})^{0.6}}$$

The corresponding growth rate  $\Omega_{KH}$  (SI unit:  $1/\text{s}$ ) of the fastest-growing disturbance is

$$\Omega_{\text{KH}} = \frac{0.34 + 0.385 \text{We}_g^{1.5}}{(1 + Z_l)(1 + 1.4T^{0.6})} \sqrt{\frac{\sigma}{\rho_p r_0^3}}$$

The characteristic breakup time  $\tau_{\text{KH}}$  (SI unit: s) can be derived from the wavelength and frequency of the fastest-growing disturbance,

$$\tau_{\text{KH}} = \frac{3.788 B_{\text{KH}} r_0}{\Lambda_{\text{KH}} \Omega_{\text{KH}}}$$

where  $B_{\text{KH}}$  is a dimensionless constant. The value of  $B_{\text{KH}}$  can vary considerably, depending on the mechanism for spray formation. Typical values are between  $\sqrt{3}$  and 60.

Kelvin-Helmholtz breakup is likely to occur when the parent droplet is sufficiently large and the time since previous breakup is comparable to or greater than  $\tau_{\text{KH}}$ . The approximate size of the parent droplet over time is computed by allocating an effective Kelvin-Helmholtz radius  $r_{\text{KH}}$  (SI unit: m) as an auxiliary dependent variable for all particles. The effective radius is initially equal to the parent droplet radius  $r_0$  and decreases over time according to the expression

$$\frac{dr_{\text{KH}}}{dt} = \frac{r_{\text{ch}} - r_{\text{KH}}}{\tau_{\text{KH}}}$$

Where  $r_{\text{ch}}$  (SI unit: m) is the child droplet radius,

$$r_{\text{ch}} = \begin{cases} B_0 \Lambda_{\text{KH}} & B_0 \Lambda_{\text{KH}} \leq r_{\text{KH}} \\ \min \left[ \left( \frac{3\pi r_{\text{KH}}^2 U_{\text{rel}}}{2\Omega_{\text{KH}}} \right)^{1/3}, \left( \frac{3\pi r_{\text{KH}}^2 U_{\text{rel}}}{4} \right)^{1/3} \right] & B_0 \Lambda_{\text{KH}} > r_{\text{KH}} \end{cases}$$

where  $B_0 = 0.61$  is a dimensionless constant.

The criterion for breakup is that, at a time step taken by the solver, there exists a positive integer  $n_{\text{ch}}$  such that

$$m_0 \geq m + n_{\text{ch}} r_{\text{ch}}$$

When this criterion is true, a number of child droplets equal to  $n_{\text{ch}}$  is released at the parent droplet's position. The new mass of the parent droplet  $m_{\text{new}}$  (SI unit: kg) is

$$m_{\text{new}} = m_0 - n_{\text{ch}} r_{\text{ch}}$$

## RAYLEIGH-TAYLOR BREAKUP

The [Rayleigh-Taylor Breakup Model](#) is a significant breakup mechanism for droplets that undergo rapid accelerations. Compared to the Kelvin-Helmholtz model, in which child droplets are continually stripped away from a liquid jet, the Rayleigh-Taylor model describes the complete annihilation of the parent droplet as it breaks into a large number of child droplets.

According to the Rayleigh-Taylor model, a droplet breaks up when the wavelength of a disturbance on the droplet's surface exceeds a critical value, which depends on the droplet's radius.

Following [Ref. 21](#), the wavelength of the fastest-growing disturbance  $\Lambda_{\text{RT}}$  (SI unit: m) is

$$\Lambda_{\text{RT}} = 2\pi \sqrt{\frac{3\sigma}{a\rho_p}}$$

The frequency of the fastest-growing wave  $\Omega_{\text{RT}}$  (SI unit: 1/s) is

$$\Omega_{\text{RT}} = \sqrt{\frac{2a}{3}} \sqrt{\frac{a\rho_p}{3\sigma}}$$

where  $a$  (SI unit:  $\text{m/s}^2$ ) is the magnitude of the droplet acceleration. The time constant for breakup  $\tau_{\text{RT}}$  (SI unit: s) is

$$\tau_{\text{RT}} = \sqrt{\frac{3}{2a}} \sqrt{\frac{3\sigma}{\rho_p a}}$$

The radius of the child droplets  $r_{\text{ch}}$  (SI unit: m) is

$$r_{\text{ch}} = \frac{C_{\text{RT}}\Lambda_{\text{RT}}}{2}$$

where  $C_{\text{RT}}$  is a dimensionless constant, with typical values ranging from 2.5 to 5.

When the Rayleigh-Taylor model is used, the time since previous Rayleigh-Taylor breakup  $t_{\text{RT}}$  (SI unit: s) is defined as an auxiliary dependent variable for all particles. Whenever  $t_{\text{RT}} > \tau_{\text{RT}}$  at the beginning of a time step taken by the solver, the parent droplet is completely broken up into child droplets. The number of child droplets  $n_{\text{ch}}$  is

$$n_{\text{ch}} = \text{floor}\left(\frac{r_0^3}{r_{\text{ch}}^3}\right)$$

The `floor` operator is used to ensure that the number of child droplets is an integer, even though the resulting droplets may have radii different from  $r_{\text{ch}}$ .

### USING MULTIPLE DROPLET BREAKUP MODELS

It is possible to include the Kelvin-Helmholtz and Rayleigh-Taylor breakup models in the same simulation. In this case, separate auxiliary dependent variables are allocated for the Kelvin-Helmholtz effective radius  $r_{\text{KH}}$  and the time since previous Rayleigh-Taylor breakup  $t_{\text{RT}}$ .

If the condition for Kelvin-Helmholtz breakup is met, then for the parent droplet and all child droplets  $t_{\text{RT}}$  is set to zero.

If the condition for Rayleigh-Taylor breakup is met, then for all child droplets the Kelvin-Helmholtz radius  $r_{\text{KH}}$  is set equal to the child droplet radius.

### *Nozzle Theory*

---

The `Nozzle` feature is used to release a spray of liquid droplets, as if they were released from a nozzle. It computes the initial size and velocity of the released droplets using the so-called Blob model (Ref. 22). This model requires the following assumptions:

- The inlet is circular and produces a round liquid jet of density  $\rho_p$  (SI unit:  $\text{kg/m}^3$ ).
- The gas in the adjacent domain is incompressible and initially quiescent,  $\mathbf{v}_{g,0} = 0$ .
- The liquid velocity  $\mathbf{v}_l$  (SI unit: m/s) is constant inside the jet.
- Both the injection pressure  $p_{\text{inj}}$  (SI unit: Pa) and the ambient pressure  $p_{\text{amb}}$  (SI unit: Pa) are constant.
- The initial parent drop radius is constant and equal to the injector radius  $r_0$  (SI unit: m).
- The maximum cone angle  $\Theta$  (SI unit: rad) is constant and stays constant as the injection progresses.



The following discussion uses several dimensionless variables. Their definitions are listed in [Table 5-1](#) in the [Droplet Breakup Theory](#) section.

---

The initial speed of the released droplets is

$$U_0 = \sqrt{\frac{2\Delta p}{\rho_p}} c_d = \frac{1}{\rho_p \pi r_0^2} \dot{m}$$

where  $c_d$  (dimensionless) is the discharge coefficient and  $\dot{m}$  (SI unit: kg/s) is the mass flow rate.

The droplets are released in a cone with maximum spray angle  $\Theta$ . The spray angle can be specified directly, or it can be estimated from Kelvin-Helmholtz instability theory:

$$\tan\left(\frac{\Theta}{2}\right) = A \frac{\Lambda \Omega}{U_0}$$

where  $\Lambda$  (SI unit: m) and  $\Omega$  (SI unit: 1/s) are, respectively, the wavelength and growth rate of the fastest-growing disturbance on the surface of a liquid jet:

$$\Lambda_{\text{KH}} = \frac{9.02 r_0 (1 + 0.45 \sqrt{Z_l}) (1 + 0.4 T^{0.7})}{(1 + 0.865 \text{We}_g^{1.67})^{0.6}}$$

$$\Omega_{\text{KH}} = \frac{0.34 + 0.385 \text{We}_g^{1.5}}{(1 + Z_l)(1 + 1.4 T^{0.6})} \sqrt{\frac{\sigma}{\rho_p r_0^3}}$$

The number multiplication factor  $n$ , activated by the **Enable macroparticles** check box in the physics interface **Additional Variables** section, is a dimensionless quantity that indicates the number of droplets that are represented by each model particle.

Suppose the range of specified release times is  $t_1, t_2, t_3, \dots, t_{N_t}$ , where  $N_t$  is the number of release times.

The droplets that are released at time  $t_i$  are assigned initial multiplication factor  $n_i$ :

$$n_i = \frac{3 \dot{m} \Delta t_i}{4 \pi N_0 \rho_p r_0^3}$$

where  $N_0$  is the number of model particles per release and the time intervals  $\Delta t_i$  are defined as



$$\Delta t_i = \begin{cases} \frac{t_2 - t_1}{2} & i = 1 \\ \frac{t_{i+1} - t_{i-1}}{2} & 1 < i < N \\ \frac{t_N - t_{N-1}}{2} & i = N \end{cases}$$

where  $N$  is the number of release times. Note that, based on the previous expressions,

$$\sum_{i=1}^{N_p} \frac{4}{3} \pi \rho_p n r_0^3 = \dot{m} (t_N - t_1)$$

where  $N_p = N \times N_0$  is the total number of released model particles. That is, the sum of the masses of all released droplets, scaled by their multiplication factors, equals the total mass released by the nozzle, assuming that the mass flow rate is constant between the initial and final release times and zero otherwise.

### *Computing Particle Temperature*

---

Select the [Compute Particle Temperature](#) check box in the physics interface **Advanced Settings** section to define an auxiliary dependent variable for particle temperature. The particle temperature  $T_p$  (SI unit: K) is then computed along each particle trajectory by integrating the first-order equation

$$m_p C_p \frac{dT_p}{dt} = Q \quad (5-4)$$

where

- $m_p$  (SI unit: kg) is the particle mass,
- $C_p$  (SI unit: J/(kg K)) is the particle specific heat capacity,
- $t$  (SI unit: s) is the time, and
- $Q$  (SI unit: W) is the total heat source at the particle surface.

The particle temperature is treated as a single value for each particle, not as a temperature distribution throughout the particle's volume. Therefore, the temperature computation is only valid when the temperature throughout the particle can be considered uniform; that is, the heat transfer resistance within the particle is

negligibly small compared to the heat transfer resistance at the surface of the particle. This is typically true for small particles with high thermal conductivity.

### CONVECTIVE HEAT LOSSES

Use the [Convective Heat Losses](#) feature to apply convective heat transfer at the surface of the particles. This feature creates the following contribution to the total heat source in [Equation 5-4](#):

$$Q = hA_p(T - T_p)$$

where

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

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

$$Bi = \frac{hL_C}{k_p}$$

where  $L_C$  (SI unit:  $m$ ) is a characteristic length, typically the ratio of particle volume to particle surface area, and  $k_p$  (SI unit:  $W/(m K)$ ) is the particle thermal conductivity. If the Biot number is very small, much less than unity, then the conductive heat transfer within the particle takes place on a much shorter time scale than convective heat transfer at the surface of the particle, so the particle temperature can be treated as a uniform value.

### RADIATIVE HEAT LOSSES

Use the [Radiative Heat Losses](#) feature to make the particles undergo radiative heat exchange with their surroundings. This feature creates the following contribution to the total heat source in [Equation 5-4](#):

$$Q = \epsilon_p \sigma A_p (T^4 - T_p^4)$$

where

- $\epsilon_p$  (dimensionless) is the particle emissivity,
- $\sigma = 5.670373 \times 10^{-8} W/(m^2 K^4)$  is the Stefan-Boltzmann constant,

- $A_p$  (SI unit:  $m^2$ ) is the particle surface area, and
- $T$  (SI unit: K) is the temperature of the enclosure or ambient surroundings.

#### USER-DEFINED HEAT SOURCE

Use the [Heat Source](#) feature to create a user-defined heat source for the particles. This feature creates a user-defined contribution to the total heat source  $Q$  in [Equation 5-4](#).

#### *Computing Particle Mass*

---

Select the [Compute Particle Mass](#) check box in the physics interface **Advanced Settings** section to allocate an auxiliary dependent variable for the mass of each particle. You can specify the **Accretion rate**  $R$  (SI unit: kg/s), the time derivative of the particle mass, in the settings window for the [Particle Properties](#) feature.

#### *Velocity of Mass Gained or Lost*

When the **Compute particle mass** check box is selected, the interpretation of Newton's second law of motion for each particle is

$$m_p \frac{d}{dt} \left( \frac{d\mathbf{q}}{dt} \right) = \mathbf{F}_t \quad (5-5)$$

instead of the more customary, momentum-conserving expression

$$\frac{d}{dt} \left( m_p \frac{d\mathbf{q}}{dt} \right) = \mathbf{F}_t \quad (5-6)$$

The physical interpretation of [Equation 5-5](#) is that the mass gained or lost by the model particle is moving with the particle's velocity, whereas the physical interpretation of [Equation 5-6](#) is that the mass gained or lost by the model particle is stationary with respect to the coordinate system in which the geometry is defined.

When **Compute particle mass** is selected, [Equation 5-5](#) is solved. To instead solve [Equation 5-6](#), so that mass gained or lost by the particle affects its velocity such that the momentum of the model particle is conserved, clear the **Compute particle mass** check box and instead either enter a time-dependent expression for the particle mass directly or express the particle mass in terms of a user-defined auxiliary dependent variable.

#### *References for the Particle Tracing for Fluid Flow Interface*

---

1. E.M. Lifshitz and L.P. Pitaevskii, *Physical Kinetics*, Butterworth, 1981.

2. E.M. Lifshitz and L.P. Pitaevskii, *Fluid Mechanics*, Butterworth, 1987.
3. T. B. Jones, *Electromechanics of Particles*, Cambridge, 1995.
4. R. Clift, J.R. Grace, and M.E. Weber, *Bubbles, Drops, and Particles*, Dover, 1978.
5. Happel, John, and Howard Brenner. Low Reynolds number hydrodynamics: with special applications to particulate media. Vol. 1. Springer Science & Business Media, 2012.
6. L. Tian, G. Ahmadi, “Particle deposition in turbulent duct flows—comparisons of different model predictions”, *Aerosol Science*, vol. 38, 2007, pp. 377-397.
7. Saffman, P. G. T, “The lift on a small sphere in a slow shear flow”, *J. Fluid Mech.* 22, no. 02 (1965), pp. 385-400.
8. Ho, B. P., and L. G. Leal, “Inertial migration of rigid spheres in two-dimensional unidirectional flows”, *J. Fluid Mech.* 65, no. 02 (1974), pp. 365-400.
9. D.J. Thomson, “Criteria for the selection of stochastic models of particle trajectories in turbulent flows”, *J. Fluid Mech.*, vol. 180, 1987, pp. 529-556.
10. A. Dehbi, “Turbulent particle dispersion in arbitrary wall-bounded geometries: A coupled CFD-Langevin-equation based approach”, *International Journal of Multiphase Flow*, vol. 34, 2008, pp. 819-828.
11. P. A. Durbin, “Stochastic Differential Equations and Turbulent Dispersion”, NASA Reference Publication 1103, 1983.
12. P. Hutchinson, G.F. Hewitt, and A.E. Dukler, “Deposition of liquid or solid dispersions from turbulent gas streams: a stochastic model”, *Chemical Engineering Science*, vol. 26, 1971, pp.419-439.
13. B. Zhao, C. Yang, X. Yang, and S. Liu, “Particle dispersion and deposition in ventilated rooms: Testing and evaluation of different Eulerian and Lagrangian models”, *Building and Environment*, vol. 43, 2008, pp. 388-397.
14. L.P. Gorkov, *Sov. Phys. Doklady*, vol. 6, p. 773, 1962.
15. H. Bruus, Lecture Notes: Microfluidics and ultrasound acoustophoreses.  
<http://www.nanotech.dtu.dk/Research/Research%20groups/TMF.aspx>
16. M. Kim and A.L. Zydney, “Effect of Electrostatic, Hydrodynamic, and Brownian Forces on Particle Trajectories and Sieving in Normal Flow Filtration,” *J. Colloid and Interface Science*, vol. 269, pp. 425–431, 2004.

17. S. Beresnev and V. Chernyak, "Thermophoresis of a Spherical Particle in a Rarefied Gas: Numerical Analysis Based on the Model Kinetic Equations," *Phys. Fluids*, vol. 7, pp. 1743-1756, 1995.
18. F. Zheng, "Thermophoresis of Spherical and Non-Spherical Particles: a Review of Theories and Experiments," *Advances in Colloid and Interface Science*, vol. 97, pp. 255-278, 2002.
19. I. Finnie, "Some Observations on the Erosion of Ductile Metals," *Wear*, vol. 19, pp. 81-90, 1972.
20. Y. Zhang, E.P. Reuterfors, B.S. McLaury, S.A. Shirazi, and E.F. Rybicki, "Comparison of computed and measured particle velocities and erosion in water and air flows," *Wear*, vol. 263, pp. 330-338, 2007.
21. M. A. Patterson and R. D. Reitz, *Modeling the Effects of Fuel Spray Characteristics on Diesel Engine Combustion and Emission*, SAE Paper 980131, 1998.
22. R. D. Reitz, *Modeling Atomization Processes in High-Pressure Vaporizing Sprays*, *Atomization and Spray Technology*, vol. 3, pp. 309-337, 1987.



# Multiphysics Interfaces


The Particle Tracing Module contains predefined multiphysics interfaces to facilitate easy setup of models with the most commonly occurring settings. This chapter describes the multiphysics interfaces that are available with the Particle Tracing Module. Two of these multiphysics interfaces can be found under the **AC/DC** branch (⚡) when adding a physics interface — Particle Field Interaction, Non-Relativistic and Particle Field Interaction, Relativistic. The Fluid Particle Interaction interface is found under the **Fluid Flow** branch (≡).


In this chapter:

- [The Particle Field Interaction, Non-Relativistic Interface](#)
- [Theory for the Particle Field Interaction, Non-Relativistic Interface](#)
- [The Particle Field Interaction, Relativistic Interface](#)
- [Theory for the Magnetic Particle Field Interaction, Relativistic Interface](#)
- [The Fluid-Particle Interaction Interface](#)
- [Theory for the Fluid-Particle Interaction Interface](#)

See [The Electromagnetics Interfaces](#) in the *COMSOL Multiphysics Reference Manual* for other AC/DC interface and feature node settings.

# The Particle Field Interaction, Non-Relativistic Interface

The **Particle Field Interaction, Non-Relativistic** () multiphysics interface combines the Charged Particle Tracing interface with the Electrostatics interface. The **Electric Particle Field Interaction** multiphysics coupling feature is added automatically. The Particle Field Interaction, Non-Relativistic interface is used to model beams of charged particles at non-relativistic speeds. The particles generate a space charge density term as they propagate through domains. The space charge density is then used as a source term in the Electrostatics interface, and the resulting electric force on the particles is computed.

When a predefined **Particle Field Interaction, Non-Relativistic** interface is added from the **AC/DC>Particle Tracing** branch () of the **Model Wizard** or **Add Physics** windows, **Electrostatics** and **Charged Particle Tracing** interfaces are added to the Model Builder. A **Multiphysics** node is also added, which automatically includes the **Electric Particle Field Interaction** multiphysics coupling.

## *On the Constituent Physics Interfaces*

The Electrostatics interface is used to compute the electric field, the electric displacement field and potential distributions in dielectrics under conditions where the electric charge distribution is explicitly prescribed. The formulation is stationary but for use together with other physics, also eigenfrequency, frequency-domain, small-signal analysis, and time-domain modeling are supported in all space dimensions. The physics interface solves Gauss' law for the electric field using the scalar electric potential as the dependent variable.

The Charged Particle Tracing interface is used to model charged particle orbits under the influence of electromagnetic forces. In addition, it can also model two-way coupling between the particles and fields. Some typical applications are particle accelerators, vacuum tubes, and ion implanters. The physics interface supports time-domain modeling only in 2D and 3D. The physics interface solves the equation of motion for charged particles subjected to electromagnetic forces.

## **SETTINGS FOR PHYSICS INTERFACES AND COUPLING FEATURES**

When physics interfaces are added using the predefined couplings — for example, **Particle Field Interaction, Non-Relativistic** — specific settings are included with the



physics interfaces and the coupling features. However, if physics interfaces are added one at a time, followed by the coupling features, these modified settings are not automatically included.

For example, if single **Electrostatics** and **Charged Particle Tracing** interfaces are added, COMSOL Multiphysics adds an empty **Multiphysics** node. You can choose from the available coupling features but the modified settings are not included.



Coupling features are available from the context menu (right-click the **Multiphysics** node) or from the **Physics** toolbar, **Multiphysics** menu.

TABLE 6-1: MODIFIED SETTINGS FOR A PARTICLE FIELD INTERACTION, NON-RELATIVISTIC INTERFACE

PHYSICS INTERFACE OR COUPLING FEATURE	MODIFIED SETTINGS (IF ANY)
Electrostatics	No changes.
Charged Particle Tracing	For the Charged Particle Tracing interface, under <b>Particle Release and Propagation</b> , the <b>Particle release specification</b> is set to <b>Specify current</b> . The <b>Electric Force</b> node is added to the Charged Particle Tracing interface. The <b>Domain Selection</b> is the same as that of the Charged Particle Tracing interface.
Electric Particle Field Interaction	The <b>Domain Selection</b> is the same as that of the participating physics interfaces. The corresponding Electrostatics and Charged Particle Tracing interfaces are preselected in the <b>Electric Particle Field Interaction</b> section.

## PHYSICS INTERFACES AND COUPLING FEATURES



Use the online help in COMSOL Multiphysics to locate and search all the documentation. All these links also work directly in COMSOL Multiphysics when using the Help system.

### *Coupling Features*

The [Electric Particle Field Interaction](#) and [Space Charge Limited Emission](#) coupling feature nodes are described in this section.

### Physics Interface Features


Physics nodes are available from the **Physics** ribbon toolbar (Windows users), **Physics** context menu (Mac 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.

- The available physics features for [The Charged Particle Tracing Interface](#) are listed in the section [Domain, Boundary, Pair, and Global Nodes for the Particle Tracing for Fluid Flow Interface](#).
- The available physics features for [The Electromagnetics Interfaces](#) are listed in the section [Domain, Boundary, Edge, Point, and Pair Nodes for the Electrostatics Interface](#) in the *COMSOL Multiphysics Reference Manual*.

### Electric Particle Field Interaction

The **Electric Particle Field Interaction** multiphysics coupling () computes the space charge density due to particles and assigns it to a dependent variable that is defined on the domain mesh elements that contain the particles. It also applies the accumulated space charge density as a source when computing the electric potential.

The computation of the space charge density is controlled by the **Particle release specification** list in the settings window for the Charged Particle Tracing interface. If **Specify release times** is selected, each particle contributes to the space charge density based on its instantaneous location. If **Specify current** is selected, each model particle is treated as representing a number of charged particles per unit time, leaving behind a contribution to the space charge density in mesh elements it has previously passed through.

#### SETTINGS

The **Label** is the default multiphysics coupling name.

The **Name** is used primarily as a scope prefix for variables defined by the coupling node. Refer to such variables in expressions using the pattern `<name>.<variable_name>`. In order to distinguish between variables belonging to different coupling nodes or 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 multiphysics coupling in the model) is `epfi1`.

#### CHARGE MULTIPLICATION FACTOR

Enter a **Charge multiplication factor**  $n$  (dimensionless). The default value is  $1 \times 10^6$ . If the **Particle release specification** in the settings window for the Charged Particle Tracing interface is set to **Specify release times**, the **Charge multiplication factor** indicates the number of real particles represented by every model particle for the purpose of computing the space charge density.

#### CONTINUATION SETTINGS

Select the **Use cumulative space charge density** check box to enable the following inputs, which are used to accelerate the convergence of models with bidirectionally coupled particle-field interactions.



The options in the **Continuation Settings** section only have an effect on the solution if the model uses the [Bidirectionally Coupled Particle Tracing](#) study step. This is a specialized study step that should be used to model bidirectionally coupled particle-field interactions, in which the contribution of the charged particles to the space charge density in the surrounding domain is sufficiently large to significantly perturb the electric field, which in turn modifies the particle trajectories.

Enter the **Number of iterations**  $\beta$  (dimensionless). The default value is 1. During the iterative solver loop set up by the [Bidirectionally Coupled Particle Tracing](#) study step, the contribution of the particles to the space charge density will be linearly ramped up for iteration numbers less than  $\beta$ . This improves the probability of stable convergence for models in which overestimation of the space charge density may lead to non-physical particle motion. This is often true when using the [Space Charge Limited Emission](#) feature, since overestimation of the space charge density in the vicinity of the cathode may cause a non-physical reversal in the direction of the initial particle velocity.

Select an option from the **Weights for subsequent iterations** list: **Uniform** (the default), **Arithmetic sequence**, or **Geometric sequence**. For **Geometric sequence** enter the **Common ratio**  $r$  (dimensionless). The default value is 1.5. The contribution of charged particles to the space charge density is treated as a cumulative average of the solutions for all iteration numbers greater than  $\beta$ , and these settings control how the cumulative average is computed.



---

For more information about the options in the **Continuation Settings** section, see [Stabilization of the Space Charge Density Calculation in Theory for the Particle Field Interaction, Non-Relativistic Interface](#).

---

### COUPLED INTERFACES

This section defines the physics involved in the multiphysics coupling. By default, the applicable physics interface is selected in the **Source** and **Destination** lists.

You can also select **None** from either list to uncouple the node from a physics interface. If the physics interface is removed from the **Model Builder** then the applicable list defaults to **None** as there is nothing to couple to.




---

If a physics interface is deleted and then added to the model again, and in order to re-establish the coupling, you need to choose the physics interface again from the **Source** or **Destination** lists. This is applicable to all multiphysics coupling nodes that would normally default to the once present physics interface. See [Multiphysics Modeling Approaches](#) in the *COMSOL Multiphysics Reference Manual*.

---

### *Space Charge Limited Emission*

---

Use the **Space Charge Limited Emission** multiphysics coupling () to model the space charge limited emission of electrons from a surface.

#### SETTINGS

The **Label** is the default multiphysics coupling name.

The **Name** is used primarily as a scope prefix for variables defined by the coupling node. Refer to such variables in expressions using the pattern <name>.<variable\_name>. In order to distinguish between variables belonging to different coupling nodes or 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 multiphysics coupling in the model) is `sc1e1`.

#### ELECTRIC POTENTIAL

Enter an **Electric potential at emission surface**  $V_0$  (SI unit: V). The default value is 0.

### NUMBER OF PARTICLES

Enter the **Number of particles per release**  $N$  (dimensionless). The default value is  $1 \times 10^3$ .

### EXTRA DIMENSION SETTINGS

Enter the **Position offset**  $o_s$  (SI unit: m). The default value is 1 mm.

Select the **Initial value of electric potential in extra dimension**—**Automatic** (the default) or **User defined**. For **User defined** enter the value for  $V_{\text{xd},0}$  (SI unit: V). The default value is 1 V.

Select the **Mesh settings**: **Automatic** (the default) or **User defined**. For **User defined** enter values for the following settings:

- **Number of elements** (dimensionless). The default value is 50.
- **Element ratio** (dimensionless). The default value is 10.

### SPACE CHARGE LIMITED EMISSION

This section defines the physics involved in the multiphysics coupling. By default, the applicable physics interface is selected in the **Source** and **Destination** lists.

You can also select **None** from either list to uncouple the node from a physics interface. If the physics interface is removed from the **Model Builder** then the applicable list defaults to **None** as there is nothing to couple to.



If a physics interface is deleted and then added to the model again, and in order to re-establish the coupling, you need to choose the physics interface again from the **Source** or **Destination** lists. This is applicable to all multiphysics coupling nodes that would normally default to the once present physics interface. See [Multiphysics Modeling Approaches](#) in the *COMSOL Multiphysics Reference Manual*.

---

# Theory for the Particle Field Interaction, Non-Relativistic Interface

The [Particle Field Interaction, Non-Relativistic Interface](#) combines charged particle tracing with electrostatics to model non-relativistic beams of charged particles that can create significant space charge density distributions in the domains that contain the particles. The space charge density may, in turn, exert a significant electric force on the particles, resulting in a bidirectional coupling between the particle trajectories and the electric potential.

If all charged particle beams are released at constant current, it is possible to significantly reduce the simulation time and computational cost by combining a time-domain calculation of the particle trajectories with a Stationary solver for the calculation of the electric potential. The two calculations can then be performed using an iterative procedure that alternates between them until a self-consistent solution is attained.

An iterative solver loop that consists of a time-dependent solver for computing particle trajectories and a stationary solver for computing all other dependent variables can be set up automatically using the [Bidirectionally Coupled Particle Tracing](#) study step, described in the *COMSOL Multiphysics Reference Manual*.

## *Electrostatics and Charged Particle Tracing Equations*

---

Under static conditions, Gauss' law can be written as a variant of Poisson's equation

$$-\nabla \cdot (\epsilon_0 \nabla V - \mathbf{P}) = \rho$$

where

- $\epsilon_0 = 8.854187817 \times 10^{-12}$  F/m is the permittivity of vacuum,
- $V$  (SI unit: V) is the electric potential,
- $\mathbf{P}$  (SI unit: C/m<sup>2</sup>) is the polarization, and
- $\rho$  (SI unit: C/m<sup>3</sup>) is the charge density.

The electric field  $\mathbf{E}$  (SI unit: V/m) can be expressed in terms of the electric potential  $V$  (SI unit: V) using the relationship

$$\mathbf{E} = -\nabla V$$

The equation of motion for a charged particle in an electromagnetic field can be written as

$$\frac{d}{dt}(m_p \mathbf{v}) = -Ze\mathbf{E} + Ze(\mathbf{v} \times \mathbf{B})$$

where

- $m_p$  (SI unit: kg) is the particle mass,
- $\mathbf{v}$  (SI unit: m/s) is the particle velocity,
- $e = 1.602176565 \times 10^{-19}$  C is the elementary charge,
- $Z$  (dimensionless) is the particle charge number, and
- $\mathbf{B}$  (SI unit: T) is the magnetic flux density.

For non-relativistic particles, the contribution of the particles to the magnetic force is assumed to be negligibly small compared to the contribution of particles to the electric force; thus, the self-magnetic field is neglected. In the next section the contribution to the electric force based on particle position is considered.

### *Space Charge Density Calculation*

---

Given an array of idealized point sources such that the position vector of the  $i$ th source is denoted  $\mathbf{q}_i$  (SI unit: m), the space charge density at position  $\mathbf{r}$  is

$$\rho(\mathbf{r}) = \sum_{i=1}^N q_i \delta(\mathbf{r} - \mathbf{q}_i) \quad (6-1)$$

where  $\delta$  is the Dirac delta function,  $q_i$  is the charge of the  $i$ th particle, and  $N$  is the total number of particles. The equation for the charge density is unusable in this form, however, for the following reasons:

- The number of particles may be extraordinarily large, making it impractical to model all of them.
- The charge density becomes infinite at the location of each idealized point source, making the calculation of the electric potential infeasible with most numerical methods.

In the following sections we discuss solutions to each of these problems.

### MODELING A REPRESENTATIVE SAMPLE OF PARTICLES

Because the number of real particles, such as ions or electrons, may be too large for every particle to be modeled individually, a practical numerical approach is to release a representative sample of model particles, allowing each model particle to make the same contribution to the space charge density as an equivalent number of real particles.

For example, instead of allocating degrees of freedom for  $10^{12}$  electrons, it will often suffice to model  $10^4$  particles, each of which has a **Charge multiplication factor** of  $10^8$ , meaning that it represents  $10^8$  electrons.

#### *Simplification for Constant-Current Beams*

If a beam of particles is released at constant current, then a full time-domain calculation of the coupled particle trajectories and electric fields may require particles to be released at a large number of time steps until a stationary solution for the electric potential is reached. This can be needlessly memory-intensive and time-consuming. An alternate approach is to release particles at time  $t = 0$  and to allow each model particle to represent a continuous stream of real particles per unit time. The number of real particles per unit time represented by each model particle is denoted the effective frequency of release,  $f_{\text{rel}}$ .

The charged particles will tend to contribute to a greater space charge contribution in regions in which they are moving slowly, causing particles that are released at successive times to be closer together. This behavior can be conveniently reproduced by defining an expression for the time derivative of the charge density, rather than the charge density itself:

$$\frac{d\rho(\mathbf{r})}{dt} = \sum_{i=1}^N f_{\text{rel},i} q_i \delta(\mathbf{r} - \mathbf{q}_i) \quad (6-2)$$



The charge density can then be computed by integrating over time, as long as sufficient time is given so that the particle trajectories can be traced completely through the modeling domain.

The frequency of release can be computed using the current and number of model particles that are specified in release feature settings. For example, for an **Inlet** node with release current magnitude  $I$  (SI unit: A) and number of particles per release  $N$  (dimensionless), the effective frequency of release is

$$f_{\text{rel}} = \frac{I}{|q|N}$$

When particle beams are assumed to have constant current, then the space charge density at the last time step includes contributions from particles at every point along their trajectories in the modeling domain. Thus, it can be applied as the space charge density term when computing the electric potential.

The treatment of particle trajectories as paths in a constant-current beam is determined by the **Particle release specification** setting in the settings window for the Charged Particle Tracing physics interface. If **Specify release times** is selected, the charge density is computed using [Equation 6-1](#) and is determined by the instantaneous positions of all model particles. Thus, it is necessary to solve for the particle trajectories and electric potential in the time domain. If **Specify current** is selected, the charge density is computed using [Equation 6-2](#) and is determined by the time history of the model particle positions.

The difference between the **Specify current** and **Specify release times** particle release specification is thus analogous to the difference between integration over **Elements and time** and integration over **Elements** as described for the [Accumulator \(Domain\)](#) node.

At this point, the effect of a bidirectional coupling between the particle trajectories and fields has not been considered. If **Specify release times** is selected from the **Particle release specification** list, this does not require special consideration because the trajectories and fields are computed simultaneously. If **Specify current** is selected, however, the trajectories and fields are computed using different study types, and an additional feedback mechanism is needed. The [Bidirectionally Coupled Particle Tracing](#) study step can be used to generate a solver sequence that does the following:

- 1 Set the space charge density contribution due to the particles to zero.
- 2 Compute the electric potential and other field variables using a Stationary solver, using the value of the space charge density computed in the previous step.

- 3 Compute the particle trajectories and the resulting space charge density in the time domain, using the field variables computed in the previous step.
- 4 Repeat steps 2 and 3 until a specified number of iterations has been reached.

If the number of iterations taken by the solver sequence is sufficiently large, the resulting solution will fully account for the bidirectional coupling between the particle trajectories and stationary fields.

#### AVOIDING INFINITELY LARGE VALUES OF THE SPACE CHARGE DENSITY

The [Electric Particle Field Interaction](#) node defines a variable for the contribution to the space charge density by particles in each mesh element. This variable is discretized using constant shape functions that are, in general, discontinuous across boundaries between elements. For a mesh element  $j$  with volume  $V_j$ , and with the **Particle release specification** set to **Specify release times**, the average space charge density  $\rho_j$  is

$$\rho_j = \frac{1}{V_j} \sum_{i=1}^N n_i q_i \int \delta(\mathbf{r} - \mathbf{q}_i) dV$$

where  $n_i$  is the charge multiplication factor of the  $i$ th model particle. The integral on the right-hand side is a volume integral over element  $j$ . The resulting charge density is the average charge density over the mesh element, which may be written more concisely as

$$\rho_j = \frac{1}{V_j} \sum_{i=1}^{N_j} n_i q_i$$

where the sum is taken over all particles that are within mesh element  $j$ .

If instead the **Particle release specification** is **Specify current**, each model particle represents a number of particles per unit time which follow along the same path, determined by the effective frequency of release  $f_{\text{rel}}$ . The space charge within the mesh element can then be expressed as the solution to the first-order equation

$$\frac{d\rho_j}{dt} = \frac{1}{V_j} \sum_{i=1}^{N_j} f_{\text{rel},i} q_i$$

## Stabilization of the Space Charge Density Calculation

---

When the **Use cumulative space charge density** check box is cleared, the contribution of the charged particles to the space charge density in the surrounding domain is overwritten at each iteration of the [Bidirectionally Coupled Particle Tracing](#) study step. If this check box is selected, then instead the space charge density contribution is computed as a cumulative average over successive iterations of the solver sequence. This often leads to more robust and consistent statistical convergence of the bidirectionally coupled model.

The process of attaining statistical convergence of a bidirectionally coupled space charge model can be separated into two steps: the ramping-up step and the cumulative averaging step.

### RAMPING-UP STEP

While the number of iterations taken by the solver sequence is less than or equal to the specified **Number of iterations**  $\beta$ , then the contribution of the particles to the space charge density is updated at each iteration by scaling the newly computed charge density contribution by a factor less than 1. This reduces the probability that the charge density will be overestimated. During the ramping-up iterations, the new value of the cumulative space charge density  $\bar{\rho}_s$  is

$$\bar{\rho}_s = \frac{\bar{\rho}_{s, \text{prev}} + \frac{\text{iter}}{\beta} \rho_s}{2}$$

where

- $\bar{\rho}_{s, \text{prev}}$  (SI unit:  $\text{C}/\text{m}^3$ ) is the stored value of the cumulative space charge density from the previous iteration,
- $\rho_s$  (SI unit:  $\text{C}/\text{m}^3$ ) is the contribution of the particles to the space charge density in the current iteration,
- $\text{iter}$  (dimensionless) is the iteration number, and
- $\beta$  (dimensionless) is the maximum number of ramping-up iterations.

### CUMULATIVE AVERAGING STEP

After the ramping-up iterations are complete ( $\text{iter} > \beta$ ), the cumulative space charge density at each subsequent iteration is

$$\bar{\rho}_s = \left( \sum_{j=1}^i w_j \right)^{-1} \left( \left( \sum_{j=1}^{i-1} w_j \right) \bar{\rho}_{s, \text{prev}} + w_i \rho_s \right)$$

$$i = \text{iter} - \beta + 1$$

where  $w_j$  is the weight of each iteration. The weights are determined by the option selected from the **Weights for subsequent iterations** list:

- For **Uniform**  $w_j = 1$ .
- For **Arithmetic sequence**  $w_j = j$ .
- For **Geometric sequence**  $w_j = r^j$  for a user-defined **Common ratio**  $r$ .

The benefit of using an **Arithmetic sequence** or **Geometric sequence** is that the early iterations have a reduced impact on the solution. Therefore, if the first few iterations have very large relative error, this early error will attenuate more quickly than when using **Uniform** weighting.

### *Theory for the Space Charge Limited Emission Node*

---

The space charge density of an electron beam can generate significant electric forces that slow or deflect the beam electrons. The self-repulsion of electrons in the beam results in a maximum allowed value for the current that is emitted from the surface. For example, the space charge limited current  $J$  in a plane-parallel vacuum diode is given by Child's Law,

$$J = \frac{4\epsilon_0}{9} \sqrt{\frac{2e}{m_e}} \frac{V^{3/2}}{d^2}$$

where

- $\epsilon_0 = 8.854187817 \times 10^{-12}$  F/m is the permittivity of vacuum,
- $e = 1.602176565 \times 10^{-19}$  C is the elementary charge,
- $m_e = 9.10938291 \times 10^{-31}$  kg is the mass of an electron,
- $V$  (SI unit: V) is the potential difference across the diode, and
- $d$  (SI unit: m) is the gap thickness.

The space charge limited current may vary depending on the shape of the geometry, but the underlying principle remains the same: excessive charge accumulation in the

domain surrounding the emitting surface will repel the released electrons and prevent them from propagating.

#### EXTRA DIMENSION DEFINITION

The assumption that electron current is space charge limited at a cathode surface imposes the condition

$$\mathbf{n} \cdot \mathbf{D} = 0$$

at the boundary. The space charge limited current that satisfies this condition is often not known in advance but can be conveniently computed by creating a fictitious surface, called the emission surface, to release particles a short distance away from the cathode. The particle velocity at the emission surface is nonzero and can be computed by solving Poisson's Equation in the thin region between the emission surface and the cathode. To avoid the explicit creation of geometric entities to represent this narrow buffer region, the gap between the emission surface and the cathode is treated as a 1D extra dimension of length  $\sigma_s$  (SI unit: m). The electric potential  $V_{xd}$  in the product space of this 1D extra dimension and the selection of the Space Charge Limited Emission node is then computed using Poisson's Equation:

$$\frac{\partial^2 V_{xd}}{\partial x^2} = -\frac{\rho}{\epsilon_0}$$

where  $x$  is the position in the extra dimension, that is, the distance from the cathode. If the current density in the gap is  $J_0$ , the charge density can be denoted

$$\rho = \frac{J_0}{v}$$

where  $v$  is the particle speed,

$$v = \sqrt{\frac{2eV_{xd}}{m_e}}$$

Thus the electric potential can be obtained by solving the equation

$$\frac{\partial^2 V_{xd}}{\partial x^2} = -\frac{J_0}{\epsilon_0 \sqrt{2eV_{xd}/m_e}}$$

for which the following boundary conditions are enforced. At the cathode ( $x=0$ ),

$$V_{\text{xd}} = 0$$

$$-\mathbf{n} \cdot \epsilon_0 \nabla V_{\text{xd}} = 0$$

At the emission surface ( $x=\sigma_s$ ),

$$V_{\text{xd}} = V$$

Where  $V$  is the electric potential that is computed in the base geometry. Although  $J_0$  is currently unknown, it can act as a Lagrange multiplier during the computation of  $V_{\text{xd}}$ , which would otherwise be overconstrained.

### STABILIZATION OF THE SPACE CHARGE LIMITED CURRENT CALCULATION

If the magnitude of the gradient of the electric potential  $V$  at the emission surface becomes too large, it is possible for the electric potential in the extra dimension  $V_{\text{xd}}$  to become negative at some point. This causes the expression for the particle velocity in the extra dimension to become complex. To prevent this occurrence, the effect of the space charge density on the electric potential  $V$  in the base geometry should not all be applied at once, but should instead be increased gradually, to allow the space charge limited current to increase in a stable manner.

The options to gradually increase the contribution to the space charge density are shown in the [Continuation Settings](#) section of the settings window for the [Electric Particle Field Interaction](#) node. The corresponding theory is outlined in the [Stabilization of the Space Charge Density Calculation](#) section.

### ADDITIONAL SETTINGS FOR SPACE CHARGE LIMITED EMISSION

The **Extra Dimension Settings** section in the settings window for the [Space Charge Limited Emission](#) node can be used to improve the accuracy and stability of the space charge limited current calculation.

The **Position offset**  $\sigma_s$  is the width of the extra dimension in which  $V_{\text{xd}}$  is computed. As  $\sigma_s$  becomes smaller relative to the size of the rest of the geometry, the aspect ratio of the extra dimension increases and the accuracy of the current calculation increases while the stability of the calculation is reduced. In general,  $\sigma_s$  should be small relative to the geometry size and to the radius of curvature of any curved boundaries in the selection of the **Space Charge Limited Emission** node.


Use the **Initial value of electric potential in extra dimension** setting to define the initial value of  $V_{xd}$ . If **Automatic** is selected, the initial value increases linearly from 0 at the cathode to 1 V at the emission surface.



Use the **Mesh settings** to control the mesh in the extra dimension. The **Automatic** settings result in a mesh that consists of 50 elements which increase according to an arithmetic sequence with an element ratio of 10. As a result, the default mesh is very fine close to the cathode and coarser at the emission surface.

To maximize the stability of the space charge limited current calculation, the following should typically be used:

- The **Number of iterations**  $\beta$  must be less than the **Number of iterations** specified in the [Bidirectionally Coupled Particle Tracing](#) study step; otherwise, the effect of the space charge density will not be fully taken into account.
- As the thickness of the extra dimension  $\sigma_g$  is reduced, it is often necessary to increase the **Number of iterations**  $\beta$ .
- The domain adjacent to the selection for the **Space Charge Limited Emission** node should be meshed using a boundary layer mesh.

# The Particle Field Interaction, Relativistic Interface

The **Particle Field Interaction, Relativistic** () multiphysics interface combines the Charged Particle Tracing, Electrostatics, and Magnetic Fields physics interfaces. The **Electric Particle Field Interaction** and **Magnetic Particle Field Interaction** multiphysics coupling features are added automatically. The Particle Field Interaction, Relativistic interface is used to model beams of relativistic charged particles. The particles generate space charge density and current density terms as they propagate through domains. The space charge density and current density are then used to compute electric and magnetic forces, respectively, which are exerted on the particles. This physics interface requires both the AC/DC Module and the Particle Tracing Module.

When a predefined **Particle Field Interaction, Relativistic** interface is added from the **AC/DC** ()>**Particle Tracing** branch () of the **Model Wizard** or **Add Physics** windows, **Electrostatics**, **Magnetic Fields**, and **Charged Particle Tracing** interfaces are added to the Model Builder. A **Multiphysics** node is also added, which automatically includes the multiphysics coupling features **Electric Particle Field Interaction** and **Magnetic Particle Field Interaction**.

## *On the Constituent Physics Interfaces*

The Electrostatics interface is used to compute the electric field, the electric displacement field and potential distributions in dielectrics under conditions where the electric charge distribution is explicitly prescribed. The formulation is stationary but for use together with other physics, also eigenfrequency, frequency-domain, small-signal analysis and time-domain modeling are supported in all space dimensions. The physics interface solves Gauss' law for the electric field using the scalar electric potential as the dependent variable.

The Magnetic Fields interface is used to compute magnetic field and induced current distributions in and around coils, conductors, and magnets. Depending on the licensed products, stationary, frequency-domain, small-signal analysis and time-domain modeling are supported in 2D and 3D. Note that the frequency- and time-domain formulations become ill-posed when approaching the static limit. You may extend the useful frequency range downward by adding a low conductivity. The physics interface solves Maxwell's equations formulated using the magnetic vector potential and, optionally for coils, the scalar electric potential as the dependent variables.



The Charged Particle Tracing interface is used to model charged particle orbits under the influence of electromagnetic forces. In addition, it can also model two-way coupling between the particles and fields. Some typical applications are particle accelerators, vacuum tubes, and ion implanters. The physics interface supports time-domain modeling only in 2D and 3D. The physics interface solves the equation of motion for charged particles subjected to electromagnetic forces.

### SETTINGS FOR PHYSICS INTERFACES AND COUPLING FEATURES

When physics interfaces are added using the predefined couplings, for example **Particle Field Interaction**, **Relativistic**, specific settings are included with the physics interfaces and the coupling features. However, if physics interfaces are added one at a time, followed by the coupling features, these modified settings are not automatically included.

For example, if single **Electrostatics**, **Magnetic Field**, and **Charged Particle Tracing** interfaces are added, COMSOL Multiphysics adds an empty **Multiphysics** node. You can choose from the available coupling features but the modified settings are not included.



Coupling features are available from the context menu (right-click the **Multiphysics** node) or from the **Physics** toolbar, **Multiphysics** menu.

TABLE 6-2: MODIFIED SETTINGS FOR A PARTICLE FIELD INTERACTION, RELATIVISTIC INTERFACE

PHYSICS INTERFACE OR COUPLING FEATURE	MODIFIED SETTINGS (IF ANY)
Electrostatics	No changes.
Magnetic Fields	No changes.
Charged Particle Tracing	For the Charged Particle Tracing interface, under <b>Particle Release and Propagation</b> , the <b>Particle release specification</b> is set to <b>Specify current</b> , and the <b>Relativistic correction</b> check box is selected.  The <b>Electric Force</b> and <b>Magnetic Force</b> nodes are added to the Charged Particle Tracing interface. The <b>Domain Selection</b> for each node is the same as that of the Charged Particle Tracing interface.

TABLE 6-2: MODIFIED SETTINGS FOR A PARTICLE FIELD INTERACTION, RELATIVISTIC INTERFACE

PHYSICS INTERFACE OR COUPLING FEATURE	MODIFIED SETTINGS (IF ANY)
Electric Particle Field Interaction	<p>The <b>Domain Selection</b> is the same as that of the participating physics interfaces.</p> <p>The corresponding Electrostatics and Charged Particle Tracing interfaces are preselected in the <b>Electric Particle Field Interaction</b> section.</p>
Magnetic Particle Field Interaction	<p>The <b>Domain Selection</b> is the same as that of the participating physics interfaces.</p> <p>The corresponding Magnetic Fields and Charged Particle Tracing interfaces are preselected in the <b>Electric Particle Field Interaction</b> section.</p>

## PHYSICS INTERFACES AND COUPLING FEATURES



Use the online help in COMSOL Multiphysics to locate and search all the documentation. All these links also work directly in COMSOL Multiphysics when using the Help system.

### *Coupling Features*

The [Magnetic Particle Field Interaction](#) coupling is described in this section. The [Electric Particle Field Interaction](#) and [Space Charge Limited Emission](#) coupling feature nodes are described for [The Particle Field Interaction, Non-Relativistic Interface](#).

### *Physics Interface Features*

Physics nodes are available from the **Physics** ribbon toolbar (Windows users), **Physics** context menu (Mac 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.


- The available physics features for [The Charged Particle Tracing Interface](#) are listed in the section [Domain, Boundary, Pair, and Global Nodes for the Particle Tracing for Fluid Flow Interface](#).

- The available physics features for [The Electromagnetics Interfaces](#) are listed in the section [Domain, Boundary, Edge, Point, and Pair Nodes for the Electrostatics Interface](#) in the *COMSOL Multiphysics Reference Manual*.
- The available physics features for [The Magnetic Fields Interface](#) are listed in the section [Domain, Boundary, Point, and Pair Nodes for the Magnetic Fields Interface](#) in the *COMSOL Multiphysics Reference Manual*.



This physics interface requires the addition of the AC/DC Module. This means that additional feature nodes are available for the Electrostatics and Magnetic Fields interfaces. These are described in the *AC/DC Module User's Guide* and the information is most easily available from the online Help found when working in COMSOL Multiphysics.

### *Magnetic Particle Field Interaction*

The **Magnetic Particle Field Interaction** multiphysics coupling () computes the current density due to the motion of charged particles and assigns the current density components to a set of dependent variables that are defined on the domain mesh elements that contain the particles. It also applies the accumulated current density as a source when computing the magnetic vector potential.

The computation of the current density is controlled by the **Particle release specification** list in the settings window for the Charged Particle Tracing interface. If **Specify release times** is selected, each particle contributes to the current density based on its instantaneous location. If **Specify current** is selected, each model particle is treated as representing a number of charged particles per unit time, leaving behind a contribution to the current density in mesh elements it has previously passed through.

#### **SETTINGS**

The **Label** is the default multiphysics coupling name.

The **Name** is used primarily as a scope prefix for variables defined by the coupling node. Refer to such variables in expressions using the pattern `<name>.<variable_name>`. In order to distinguish between variables belonging to different coupling nodes or 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 multiphysics coupling in the model) is `mpf1`.

## MAGNETIC PARTICLE FIELD INTERACTION

This section defines the physics involved in the multiphysics coupling. By default, the applicable physics interface is selected in the **Source** and **Destination** lists.

You can also select **None** from either list to uncouple the node from a physics interface. If the physics interface is removed from the **Model Builder** then the applicable list defaults to **None** as there is nothing to couple to.



If a physics interface is deleted and then added to the model again, and in order to re-establish the coupling, you need to choose the physics interface again from the **Source** or **Destination** lists. This is applicable to all multiphysics coupling nodes that would normally default to the once present physics interface. See [Multiphysics Modeling Approaches](#) in the *COMSOL Multiphysics Reference Manual*.

---

# Theory for the Magnetic Particle Field Interaction, Relativistic Interface

The [Particle Field Interaction, Relativistic Interface](#) combines the Charged Particle Tracing, Electrostatics, and Magnetic Fields interfaces to model relativistic beams of charged particles that can create significant space charge density and current density distributions in the domains that contain the particles. The space charge density may, in turn, exert a significant electric force on the particles, whereas the current density contributes to the magnetic field in the surrounding domains, exerting a magnetic force on the particles.

If all charged particle beams are released at constant current, it is possible to significantly reduce the simulation time and computational cost by combining a time-domain calculation of the particle trajectories with a Stationary solver for the calculation of the electric potential and magnetic vector potential. The two calculations can then be performed using an iterative procedure that alternates between them until a self-consistent solution is attained.

An iterative solver loop that consists of a time-dependent solver for computing particle trajectories and a stationary solver for computing all other dependent variables can be set up automatically using the [Bidirectionally Coupled Particle Tracing](#) study step, described in the *COMSOL Multiphysics Reference Manual*.

## *Electrostatics and Charged Particle Tracing Equations*

---

Under static conditions, Gauss' law can be written as a variant of Poisson's equation

$$-\nabla \cdot (\epsilon_0 \nabla V - \mathbf{P}) = \rho$$

where

- $\epsilon_0 = 8.854187817 \times 10^{-12}$  F/m is the permittivity of vacuum,
- $V$  (SI unit: V) is the electric potential,
- $\mathbf{P}$  (SI unit: C/m<sup>2</sup>) is the polarization, and
- $\rho$  (SI unit: C/m<sup>3</sup>) is the charge density.

The electric field  $\mathbf{E}$  (SI unit: m/s) can be expressed in terms of the electric potential  $V$  (SI unit: V) using the relationship

$$\mathbf{E} = -\nabla V$$

The equation for the magnetic vector potential  $\mathbf{A}$  (SI unit: Wb/m) may be written as

$$\sigma \frac{\partial \mathbf{A}}{\partial t} + \nabla \times \mathbf{H} = \mathbf{J}_e$$

$$\mathbf{B} = \nabla \times \mathbf{A}$$

where  $\sigma$  (SI unit: S/m) is the electrical conductivity,  $\mathbf{H}$  (SI unit: A/m) is the magnetic field,  $\mathbf{B}$  (SI unit: T) is the magnetic flux density, and  $\mathbf{J}_e$  (SI unit: A/m<sup>2</sup>) is the external current density. The constitutive relationship between  $\mathbf{B}$  and  $\mathbf{H}$  can be written

$$\mathbf{B} = \mu_0(\mathbf{H} + \mathbf{M})$$

where  $\mu_0$  (SI unit: H/m) is the permeability of free space and  $\mathbf{M}$  (SI unit: A/m) is the magnetization.

The equation of motion for a charged particle in an electromagnetic field can be written as

$$\frac{d}{dt}(m_p \mathbf{v}) = -Ze\mathbf{E} + Ze(\mathbf{v} \times \mathbf{B})$$

where

- $m_p$  (SI unit: kg) is the particle mass,
- $\mathbf{v}$  (SI unit: m/s) is the particle velocity,
- $e = 1.602176565 \times 10^{-19}$  C is the elementary charge,
- $Z$  (dimensionless) is the particle charge number, and
- $\mathbf{B}$  (SI unit: T) is the magnetic flux density.

For relativistic particles, the particle mass is expressed in terms of the rest mass  $m_r$  (SI unit: kg):

$$m_p = \frac{m_r}{\sqrt{1 - \mathbf{v} \cdot \mathbf{v}/c^2}}$$

where  $c = 2.99792458 \times 10^8$  m/s is the speed of light in a vacuum. For non-relativistic particles, the particle mass can be replaced by the rest mass and the self-imposed

magnetic force of the beam is often negligibly small compared to the electric force, allowing [The Particle Field Interaction, Non-Relativistic Interface](#) to be used effectively.

The calculation of the space charge density term and the resulting effect on the electric force is explained in [Space Charge Density Calculation in Theory for the Particle Field Interaction, Non-Relativistic Interface](#). The contribution of particle motion to the magnetic force is considered in the following section.

### *Current Density Calculation*

---

Given an array of idealized point sources such that the position vector of the  $i$ th source is denoted  $\mathbf{q}_i$  (SI unit: m), the contribution to the current density by particles  $\mathbf{j}_s$  at position  $\mathbf{r}$  is

$$\mathbf{j}_s(\mathbf{r}) = \sum_{i=1}^N Z_i e \mathbf{v}_i \delta(\mathbf{r} - \mathbf{q}_i) \quad (6-3)$$

where

- $\delta$  is the Dirac delta function,
- $Z_i$  (dimensionless) is the charge number of the  $i$ th particle,
- $e = 1.602176565 \times 10^{-19}$  C is the elementary charge,
- $\mathbf{v}_i$  (SI unit: m/s) is the velocity of the  $i$ th particle, and
- $N$  (dimensionless) is the total number of particles.

The equation for the current density is unusable in this form, however, because the number of particles involved may be extraordinarily large, and because the transfer of information from the point particles to degrees of freedom defined on a finite element mesh introduces some discretization error. In the following sections we discuss solutions to these problems.

### **MODELING A REPRESENTATIVE SAMPLE OF PARTICLES**

Because the number of real particles, such as ions or electrons, may be too large for every particle to be modeled individually, a practical numerical approach is to release a representative sample of model particles, allowing each model particle to make the same contribution to the current density as an equivalent number of real particles.

For example, instead of allocating degrees of freedom for  $10^{12}$  electrons, it will often suffice to model  $10^4$  particles, each of which has a **Charge multiplication factor** of  $10^8$ , meaning that it represents  $10^8$  electrons.

#### *Simplification for Constant-Current Beams*

If a beam of particles is released at constant current, then a full time-domain calculation of the coupled particle trajectories and electric and magnetic fields may require particles to be released at a large number of time steps until a stationary solution for the electric potential and magnetic vector potential is reached. This can be needlessly memory-intensive and time-consuming. An alternate approach is to release particles at time  $t = 0$  and to allow each model particle to represent a continuous stream of real particles per unit time. The number of real particles per unit time represented by each model particle is denoted the effective frequency of release,  $f_{\text{rel}}$ .

The charged particles will contribute to the current density along their entire trajectories, not just at their instantaneous positions. This behavior can be conveniently reproduced by defining an expression for the time derivative of the current density, rather than the current density itself:

$$\frac{d\mathbf{j}_s(\mathbf{r})}{dt} = \sum_{i=1}^N f_{\text{rel},i} q_i \mathbf{v}_i \delta(\mathbf{r} - \mathbf{q}_i) \quad (6-4)$$

The current density can then be computed by integrating over time, as long as sufficient time is given so that the particle trajectories can be traced completely through the modeling domain.

The frequency of release can be computed using the current and number of model particles that are specified in release feature settings. For example, for an **Inlet** node with release current magnitude  $I$  (SI unit: A) and number of particles per release  $N$  (dimensionless), the effective frequency of release is

$$f_{\text{rel}} = \frac{I}{|q|N}$$

When particle beams are assumed to have constant current, then the current density at the last time step includes contributions from particles at every point along their trajectories in the modeling domain. Thus, it can be applied as the current density term when computing the magnetic vector potential.

The treatment of particle beams as constant-current beams is determined by the **Particle release specification** list in the settings window for the Charged Particle Tracing



physics interface. If **Specify release times** is selected, the charge density is computed using Equation 6-3 and is determined by the instantaneous positions of all model particles. Thus, it is necessary to solve for the particle trajectories, electric potential, and magnetic vector potential in the time domain. If **Specify current** is selected, the current density is computed using Equation 6-4 and is determined by the time history of the model particle positions.

The difference between the **Specify current** and **Specify release times** particle release specifications is thus analogous to the difference between integration over **Elements and time** and integration over **Elements** as described for the **Accumulator (Domain)** node.

At this point, the effect of a bidirectional coupling between the particle trajectories and fields has not been considered. If **Specify release times** is selected from the **Particle release specification** list, this does not require special consideration because the trajectories and fields are computed simultaneously. If **Specify current** is selected, however, the trajectories and fields are computed using different study types, and an additional feedback mechanism is needed. The **Bidirectionally Coupled Particle Tracing** study step generates a solver sequence that does the following:

- 1 Set the current density contribution due to the particles to zero.
- 2 Compute the magnetic vector potential and other field variables using a Stationary solver, using the value of the current density computed in the previous step.
- 3 Compute the particle trajectories and the resulting current density in the time domain, using the field variables computed in the previous step.
- 4 Repeat steps 2 and 3 until a specified number of iterations has been reached.

Given a sufficient number of iterations, the resulting solution will fully account for the bidirectional coupling between the particle trajectories and stationary fields.

#### AVOIDING INFINITELY LARGE VALUES OF THE CURRENT DENSITY

The **Magnetic Particle Field Interaction** node defines a variable for each component of the contribution to the current density by particles in each mesh element. This variable is discretized using constant shape functions. For a mesh element  $j$  with volume  $V_j$ , and with the **Particle release specification** set to **Specify release times**, the average current density  $\rho_j$  is

$$\mathbf{j}_{s,j} = \frac{1}{V_j} \sum_{i=1}^N n_i q_i \mathbf{v}_i \int \delta(\mathbf{r} - \mathbf{q}_i) dV$$

where  $n_i$  (dimensionless) is the charge multiplication factor of the  $i$ th model particle. The integral on the right-hand side is a volume integral over element  $j$ . The resulting current density is the average current density over the mesh element, which may be written as


$$\mathbf{j}_{s,j} = \frac{1}{V_j} \sum_{i=1}^{N_j} n_i q_i \mathbf{v}_i$$



where the sum is taken over all particles that are within mesh element  $j$ .

If instead the **Particle release specification** is **Specify current**, each model particle represents a number of particles per unit time which follow along the same path, determined by the effective frequency of release  $f_{\text{rel}}$ . Then the time derivative of the current density can be expressed as

$$\frac{d\mathbf{j}_{s,j}}{dt} = \frac{1}{V_j} \sum_{i=1}^{N_j} f_{\text{rel},i} q_i \mathbf{v}_i$$

# The Fluid-Particle Interaction Interface

The **Fluid-Particle Interaction** () multiphysics interface combines the Particle Tracing for Fluid Flow interface coupled with the Laminar Flow interface. **The Fluid Particle Interaction** multiphysics coupling feature is added automatically. The Fluid Particle Interaction interface is used to model the motion of particles in a fluid. As the particles are accelerated or decelerated by the drag force exerted by the fluid, the corresponding reaction force is applied to the fluid.

When a predefined **Fluid-Particle Interaction** interface is added from the **Fluid Flow** ()>**Particle Tracing** branch () of the **Model Wizard** or **Add Physics** windows, **Laminar Flow** and **Particle Tracing for Fluid Flow** interfaces are added to the Model Builder. A **Multiphysics** node is also added, which automatically includes the multiphysics coupling feature **Fluid-Particle Interaction**.

## *On the Constituent Physics Interfaces*

The equations solved by the Laminar Flow interface are the Navier-Stokes equations for conservation of momentum and the continuity equation for conservation of mass. A **Fluid Model** is active by default on all the interface selection. The flow interface domain selection may be edited if the model contains solid domains.

The Particle Tracing for Fluid Flow interface is used to compute the motion of particles in a background fluid. Particle motion can be driven by drag, gravity, and electric, magnetic, and acoustophoretic forces. User-defined forces can be added. It is also possible to compute the particle mass and temperature.

## **SETTINGS FOR PHYSICS INTERFACES AND COUPLING FEATURES**

When physics interfaces are added using the predefined couplings specific settings are included with the physics interfaces and the coupling features. However, if physics interfaces are added one at a time, followed by the coupling features, these modified settings are not automatically included.

For example, if single **Laminar Flow** and **Particle Tracing for Fluid Flow** interfaces are added, COMSOL adds an empty **Multiphysics** node. You can choose from the available coupling features but the modified settings are not included.



Coupling features are available from the context menu (right-click the **Multiphysics** node) or from the **Physics** toolbar, **Multiphysics** menu.

TABLE 6-3: MODIFIED SETTINGS FOR A FLUID-PARTICLE INTERACTION INTERFACE

PHYSICS INTERFACE OR COUPLING FEATURE	MODIFIED SETTINGS (IF ANY)
Laminar Flow	No changes.
Particle Tracing for Fluid Flow	For the Particle Tracing for Fluid Flow interface, under <b>Particle Release and Propagation</b> , the <b>Particle release specification</b> is set to <b>Specify mass flow rate</b> . The <b>Drag Force</b> node is added to the Particle Tracing for Fluid Flow interface. The <b>Domain Selection</b> is the same as that of the Particle Tracing for Fluid Flow interface.
Fluid-Particle Interaction	The <b>Domain Selection</b> is the same as that of the participating physics interfaces. The corresponding Laminar Flow and Particle Tracing for Fluid Flow interfaces are preselected in the <b>Fluid-Particle Interaction</b> section.

## PHYSICS INTERFACES AND COUPLING FEATURES



Use the online help in COMSOL Multiphysics to locate and search all the documentation. All these links also work directly in COMSOL Multiphysics when using the Help system.

### *Coupling Features*

The [Fluid-Particle Interaction](#) coupling feature node is described in this section.

### *Physics Interface Features*

Physics nodes are available from the **Physics** ribbon toolbar (Windows users), **Physics** context menu (Mac 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.

- The available physics features for [The Particle Tracing for Fluid Flow Interface](#) are listed in the section [Domain, Boundary, Pair, and Global Nodes for the Particle Tracing for Fluid Flow Interface](#).
- The available physics features for [The Single-Phase Flow, Laminar Flow Interface](#) are listed in the section [Domain, Boundary, Pair, and Point Nodes for Single-Phase Flow](#) in the *COMSOL Multiphysics Reference Manual*.

## *Fluid-Particle Interaction*

---

The **Fluid-Particle Interaction** multiphysics coupling () computes a volume force that is equal in magnitude and opposite in direction to the total drag force exerted on particles in each mesh element in the selected domains. This volume force contributes to the total force acting on the fluid in the Laminar Flow interface.

### SETTINGS

The **Label** is the default multiphysics coupling name.

The **Name** is used primarily as a scope prefix for variables defined by the coupling node. Refer to such variables in expressions using the pattern `<name>.<variable_name>`. In order to distinguish between variables belonging to different coupling nodes or 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 multiphysics coupling in the model) is `fpi1`.

### FORCE MULTIPLICATION FACTOR

Use this section to specify a proportionality factor to multiply by the change in particle momentum when computing the volume force.

Select an option from the **Force multiplication factor specification** list — **From physics** (the default) or **User defined**.

If **Specify mass flow rate** is selected from the **Particle Properties** section in the Settings window for [The Particle Tracing for Fluid Flow Interface](#), this section has no effect because the proportionality factor is based on the mass flow rate of particles, which is specified in the particle release features. Otherwise, the following conditions apply:

- If **From physics** is selected and the **Enable macroparticles** check box is cleared in the physics interface **Additional Variables** section, the multiplication factor is 1.
- If **From physics** is selected and the **Enable macroparticles** check box is selected in the physics interface **Additional Variables** section, the multiplication factor is based on the

auxiliary dependent variable for the multiplication factor, which is typically specified in release feature settings. If the physics interface has name `fpt`, this variable has name `fpt.nn`.

- If **User defined** is selected, enter a value or expression for the **Force multiplication factor**  $n$  (dimensionless). The default is 1.

### COUPLED INTERFACES

This section defines the physics involved in the multiphysics coupling. By default, the applicable physics interface is selected in the **Source** and **Destination** lists.

You can also select **None** from either list to uncouple the node from a physics interface. If the physics interface is removed from the **Model Builder** then the applicable list defaults to **None** as there is nothing to couple to.



If a physics interface is deleted and then added to the model again, and in order to re-establish the coupling, you need to choose the physics interface again from the **Source** or **Destination** lists. This is applicable to all multiphysics coupling nodes that would normally default to the once present physics interface. See [Multiphysics Modeling Approaches](#) in the *COMSOL Multiphysics Reference Manual*.

---

# Theory for the Fluid-Particle Interaction Interface

The [Fluid-Particle Interaction Interface](#) combines the Particle Tracing for Fluid Flow and Laminar Flow interfaces to model the motion of particles in a fluid in which the acceleration or deceleration of particles creates a significant volume force that affects the motion of the fluid. The [Fluid-Particle Interaction](#) node computes a volume force that is equal in magnitude and opposite in direction to the total drag force that the fluid exerts on particles.

If particles are released into the fluid at a constant mass flow rate, it is possible to significantly reduce the simulation time and computational cost by combining a time-domain calculation of the particle trajectories with a Stationary solver for the calculation of the pressure and fluid velocity. The two calculations can then be performed using an iterative procedure that alternates between them until a self-consistent solution is attained.

An iterative solver loop that consists of a time-dependent solver for computing particle trajectories and a stationary solver for computing all other dependent variables can be set up automatically using the [Bidirectionally Coupled Particle Tracing](#) study step, described in the *COMSOL Multiphysics Reference Manual*.

## *Laminar Flow and Particle Tracing Equations*

---

For an incompressible single-phase fluid in the laminar flow regime, the Navier-Stokes equations can be reduced to the following:

$$\rho \frac{\partial \mathbf{u}}{\partial t} + \rho(\mathbf{u} \cdot \nabla)\mathbf{u} = \nabla \cdot [-p\mathbf{I} + \mu(\nabla\mathbf{u} + (\nabla\mathbf{u})^T)] + \mathbf{F}$$

$$\rho \nabla \cdot \mathbf{u} = 0$$

where

- $\mathbf{u}$  (SI unit: m/s) is the fluid velocity,
- $p$  (SI unit: Pa) is the pressure,
- $\rho$  (SI unit: kg/m<sup>3</sup>) is the density,

- $\mu$  (SI unit: Pa·s) is the dynamic viscosity, and
- $\mathbf{F}$  (SI unit: N) is the total volume force.

In the following, let  $\mathbf{F}_V$  denote the contribution of particle motion to the total volume force acting on the fluid.

The equation of motion of particles in the fluid can be written as

$$\frac{d}{dt}(m_p \mathbf{v}) = \mathbf{F}_D + \mathbf{F}_g + \mathbf{F}_{\text{ext}}$$

where

- $m_p$  is the particle mass (SI unit: kg)
- $\mathbf{v}$  is the velocity of the particle (SI unit: m/s)
- $\mathbf{F}_D$  is the drag force (SI unit: N),
- $\mathbf{F}_g$  is the gravitational force (SI unit: N), and
- $\mathbf{F}_{\text{ext}}$  is any other external force (SI unit: N).

Several different expressions for the drag force are available; these are described in the section [Particle Motion in a Fluid](#) in [Theory for the Particle Tracing for Fluid Flow Interface](#).

### *Volume Force Calculation*

---

Given an array of idealized point masses such that the position vector of the  $i$ th particle is denoted  $\mathbf{q}_i$  (SI unit: m), the volume force at position  $\mathbf{r}$  is

$$\mathbf{F}_V(\mathbf{r}) = - \sum_{i=1}^N \mathbf{F}_{D,i} \delta(\mathbf{r} - \mathbf{q}_i) \quad (6-5)$$

where  $\delta$  is the Dirac delta function,  $\mathbf{F}_{D,i}$  is the drag force exerted on the  $i$ th particle, and  $N$  is the total number of particles. The equation for the volume force is unusable in this form, however, for the following reasons:

- The number of particles may be very large, making it impractical to model all of them.
- The magnitude of the volume force becomes infinite at the location of each idealized point mass, making the calculation of the volume force infeasible with most numerical methods.



In the following sections we discuss solutions to each of these problems.

### MODELING A REPRESENTATIVE SAMPLE OF PARTICLES

Because the number of real particles may be too large for every particle to be modeled individually, a practical numerical approach is to release a representative sample of model particles, allowing each model particle to make the same contribution to the volume force as an equivalent number of real particles.

For example, instead of allocating degrees of freedom for  $10^7$  small particles, it will often suffice to model  $10^4$  particles, each of which has a **Force multiplication factor** of  $10^3$ , meaning that it exerts a volume force that is  $10^3$  times greater in magnitude than the drag force that acts on it.

#### *Simplification for Constant Mass Flow Rate*

If particles are released into the fluid at a constant mass flow rate, then a full time-domain calculation of the coupled particle trajectories and field variables may require particles to be released at a large number of time steps until a stationary solution is reached. The calculation of the fluid velocity and pressure at each time step can be needlessly memory-intensive and time-consuming. An alternate approach is to release particles at time  $t = 0$  and to allow each model particle to represent a continuous stream of real particles per unit time. The number of real particles per unit time represented by each model particle is denoted the effective frequency of release,  $f_{\text{rel}}$ .

The behavior of a continuous stream of particles can be conveniently modeled by defining an expression for the time derivative of the volume force, rather than the volume force itself:

$$\frac{d\mathbf{F}_V(\mathbf{r})}{dt} = - \sum_{i=1}^N f_{\text{rel},i} \mathbf{F}_{D,i} \delta(\mathbf{r} - \mathbf{q}_i) \quad (6-6)$$

The charge density can then be computed by integrating over time, as long as sufficient time is given so that the particle trajectories can be traced completely through the modeling domain.

The frequency of release can be computed using the current and number of model particles that are specified in release feature settings. For example, for an **Inlet** node with release current magnitude  $\dot{m}$  (SI unit: kg/s) and number of particles per release  $N$  (dimensionless), the effective frequency of release is

$$f_{\text{rel}} = \frac{\dot{m}}{m_p N}$$

When the mass flow rate can be assumed to be constant, then the volume force at the last time step includes contributions from particles at every point along their trajectories in the modeling domain. Thus, it can be applied as the volume force term when computing the fluid pressure and velocity.

The treatment of the constant mass flow rate is determined by the **Particle release specification** list in the settings window for the Particle Tracing for Fluid Flow interface. If **Specify release times** is selected, the volume force is computed using [Equation 6-5](#) and is determined by the instantaneous positions of all model particles. Thus, it is necessary to solve for the particle trajectories, fluid velocity, and pressure in the time domain. If **Specify mass flow rate** is selected, the volume force is computed using [Equation 6-6](#) and is determined by the time history of the model particle positions.

The difference between the **Specify mass flow rate** and **Specify release times** option in the **Particle release specification** list is thus analogous to the difference between integration over **Elements and time** and integration over **Elements** as described for the [Accumulator \(Domain\)](#) node.

At this point, the effect of a bidirectional coupling between the particle trajectories and fields has not been considered. For the **Specify release times** option, this does not require special consideration because the trajectories and fields are computed simultaneously. For the **Specify mass flow rate** option, however, the trajectories and fields are computed using different study types, and an additional feedback mechanism is needed. The [Bidirectionally Coupled Particle Tracing](#) study step can be used to generate a solver sequence that does the following:

- 1 Set the volume force exerted by the particles on the fluid to zero.
- 2 Compute the fluid velocity and pressure using a Stationary solver, using the value of the volume force computed in the previous step.
- 3 Compute the particle trajectories and the resulting volume force in the time domain, using the field variables computed in the previous step.
- 4 Repeat steps 2 and 3 until a specified number of iterations has been reached.

If the number of iterations taken by the solver sequence is sufficiently large, the resulting solution will fully account for the bidirectional coupling between the particle trajectories and stationary fields.

### AVOIDING INFINITELY LARGE VALUES OF THE VOLUME FORCE

The [Fluid-Particle Interaction](#) node defines variables for each component of the volume force exerted by particles on the surrounding fluid. These variables are discretized using constant shape functions that are, in general, discontinuous across boundaries between elements. For a mesh element  $j$  with volume  $V_j$ , and with the **Particle release specification** set to **Specify release times**, the average volume force  $\mathbf{F}_{V,j}$  is

$$\mathbf{F}_{V,j} = -\frac{1}{V_j} \sum_{i=1}^N n_i \mathbf{F}_{D,i} \int \delta(\mathbf{r} - \mathbf{q}_i) dV$$

where  $n_i$  is the force multiplication factor of the  $i$ th model particle. The integral on the right-hand side is a volume integral over element  $j$ . The resulting volume force is the average volume force over the mesh element, which may be written more concisely as

$$\mathbf{F}_{V,j} = -\frac{1}{V_j} \sum_{i=1}^{N_j} n_i \mathbf{F}_{D,i}$$

where the sum is taken over all particles that are within mesh element  $j$ .

If instead the **Particle release specification** is **Specify mass flow rate**, each model particle represents a number of particles per unit time which follow along the same path, determined by the effective frequency of release  $f_{\text{rel}}$ . The volume force within the mesh element can then be expressed as the solution to the first-order equation

$$\frac{d\mathbf{F}_{V,j}}{dt} = -\frac{1}{V_j} \sum_{i=1}^{N_j} f_{\text{rel},i} \mathbf{F}_{D,i}$$



## Glossary

This [Glossary of Terms](#) contains modeling terms in a particle tracing context. For mathematical terms as well as geometry and CAD terms specific to the COMSOL Multiphysics® software and documentation, see the glossary in the *COMSOL Multiphysics Reference Manual*. For references to more information about a term, see the index.

# Glossary of Terms

**Brownian motion** The random drifting of particles suspended in a fluid.

**Coulomb force** The force between charged particles which is inversely proportional to the square of the distance between the particles.

**dispersed flow** A fluid-particle system where particle-fluid and particle-particle interactions need to be accounted for in the model.

**Hamiltonian** A convenient way of describing how a system of particles interact with surrounding fields. The Hamiltonian is usually defined as the sum of the kinetic and potential energy.

**Lagrangian** A convenient way of describing how a system of particles interact with surrounding fields. The Lagrangian is usually defined as the kinetic energy minus the potential energy.

**Maxwellian velocity distribution** Describes the probability that a velocity is near a given value as a function of the temperature of the system.

**residence time** The average amount of time that a particle spends in a particular system. The residence time can be computed by adding Auxiliary dependent variables.

**space charge effects** When the number density of charged particles is sufficiently high, they can affect the field in which they are placed. This is often referred to as a “space charge effect.”

**sparse flow** A particle laden flow is described as a sparse flow when the particles have no appreciable effect on the motion of the fluid.

**transmission probability** The probability that a particle transmits from a given selected boundary or domain to another.

# I n d e x

- A**
  - accumulator (node), boundaries 71
  - accumulator (node), domains 82
  - accumulator theory, boundaries 120
  - accumulator theory, domains 119
  - accuracy order, of time stepping 64
  - acoustophoretic force (node) 203
  - acoustophoretic force, theory 243
  - Application Libraries window 16
  - application library examples
    - Brownian force 202
    - charged particle tracing 126
    - for-end for loop 48
    - mathematical particle tracing 65
    - Monte Carlo modeling 45
    - newtonian, first order 33
    - particle field interaction 22
    - particle release feature 39
    - particle tracing 12
    - particle tracing for fluid flow 192
    - particle-particle interactions 86
    - sparse flow 24, 212
    - thermionic emission 153
    - transmission probability 40
  - attachment (node) 132
  - auxiliary dependent variable 70, 116
  - auxiliary dependent variable (node) 100
  - auxiliary dependent variables, initializing 50
  - azimuthal particle velocity 125
- B**
  - bndenv operator 12
  - bounce, wall condition 67, 117
  - boundary conditions
    - theory 116
  - boundary load (node) 213
  - Brownian force (node) 200
  - Brownian force, theory 240
- C**
  - centrifugal force 80, 110
  - charged particle tracing 21
  - charged particle tracing interface 124
    - theory 162
  - collision diameter 85
  - collisions (node) 128
  - common settings 14
  - continuous random walk 231
  - convective heat losses (node) 215
  - Coriolis force 80, 110
  - Coulomb force 22, 113
  - Coupling Operators 42
  - current density (node) 159
- D**
  - degrees of freedom, out-of-plane 125
  - dense flow 23
  - dielectrophoretic force (node) 205
  - dielectrophoretic force, theory 242
  - diffuse scattering, wall condition 68, 117
  - dilute flow 24
  - disappear, wall condition 68, 116
  - discrete random walk 236
  - dispersed flow 25
  - documentation 15
  - drag force (node) 196
  - drag force, theory 223
  - droplet breakup 216, 251
  - droplet sprays in fluid flow interface 222
- E**
  - elastic (node) 130
  - electric force (node) 154, 205
  - electric force, theory 241
  - electric particle field interaction (node),
    - multiphysics 266
  - emailing COMSOL 17
  - env operator 12
  - erosion (node) 211

- etch (node) 160
- Euler force 80, 110
- excitation (node) 132
- F**
  - filtering particles to view 53
  - fluid flow, particle tracing for 23
  - fluid-particle interaction (node), multi-physics 293
  - fluid-particle interaction interface 291
  - force (node) 79
  - forces
    - on particles in fluids 223
  - freeze, wall condition 67, 117
  - friction force (node) 137
- G**
  - general reflection, wall condition 68, 118
  - gravity force (node) 202
  - gravity force, theory 230
- H**
  - Hamiltonian 78
  - Hamiltonian formulation 108
  - heat source (node) 160, 216
- I**
  - initial conditions, particle position 110
  - initial velocity of particles 111
  - initializing 50
  - inlet (node)
    - particle tracing 94
  - internet resources 15
  - ionization (node) 133
  - ionization loss (node), particle-matter interactions 139
- K**
  - Kelvin-Helmholtz breakup model 217, 252
  - knowledge base, COMSOL 17
- L**
  - Lagrangian 78
  - Lagrangian formulation 108
  - Lennard-Jones force 85
  - Lennard-Jones potential 114
  - lift force (node) 199
- lift force, theory 237
- London dispersion force 115
- Lorentz force 163
- M**
  - magnetic force (node) 156, 205
  - magnetic force, theory 242
  - magnetic particle field interaction (node), multiphysics 283
  - magnetophoretic force, theory 248
  - mass deposition (node) 212
  - mass flux (node) 213
  - massless formulation 78
  - massless particle 109
  - mathematical particle tracing interface 60
    - theory 106
  - Maxwellian initial velocity distribution 111
  - mixed diffuse and specular reflection, wall condition 68, 118
  - Monte Carlo modeling 44
  - Monte Carlo simulation 201
  - MPH-files 16
  - multiphysics
    - electric particle field interaction 266
    - magnetic particle field interaction 283
    - space charge limited emission 268
  - multiphysics coupling 293
- N**
  - Newton's second law 106, 223
  - nodes, common settings 14
  - nonlocal accumulator (node) 98
  - nonresonant charge exchange (node) 135
  - Nozzle 221
  - nozzle 218, 255
  - nozzle domain 221
  - nuclear stopping (node), particle-matter interactions 140
- O**
  - operators



- env and bndenv 12
- outlet (node)
  - particle tracing 99
- out-of-plane degrees of freedom 125
- override properties (node) 103
- P**
  - particle beam (node) 141
  - particle continuity (node) 100
  - particle field interaction, non-relativistic
    - interface 264
  - particle field interaction, relativistic interface 280
  - particle index 39
  - particle mass 77, 153
  - particle motion in fluids 223
  - particle properties (node) 77, 153, 194
  - particle release time 39
  - particle tracing for fluid flow interface
    - 188, 222
    - theory 223
  - particle velocity 78
  - particle-matter interactions (node) 138
  - particle-particle interaction (node) 84
  - particle-particle interactions 113
  - pass through, wall condition 68
  - Péclet number 27
  - periodic condition (node) 70
  - physics interfaces, common settings 14
- R**
  - radiative heat losses (node) 215
  - Rayleigh-Taylor breakup model 218, 254
  - release (node) 86
    - release from data file (node) 103
    - release from edge (node) 101
    - release from grid (node) 102
    - release from point (node) 101
  - residence time 48
  - resonant charge exchange (node) 134
  - rotating frame (node) 80
  - rotating frame, theory 109
- S**
  - Saffman lift force 237
  - secondary emission (node) 73
  - shell (node) 207
  - space charge density calculation (node)
    - 158
  - space charge effects 21
  - space charge limited emission (node),
    - multiphysics 268
  - sparse flow 23
  - standard settings 14
  - stick, wall condition 68, 117
  - surface charge density (node) 159
  - symmetry (node)
    - particle tracing 154, 195
- T**
  - technical support, COMSOL 17
  - theory
    - charged particle tracing 162
    - mathematical particle tracing 106
    - particle tracing for fluid flow interface
      - 223
  - thermionic emission (node) 151
  - thermionic emission, theory 178
  - thermophoretic force (node) 209
  - thermophoretic force, theory 244
  - total number of particles 40
  - transmission probability 40
- U**
  - user defined (node), collisions 136
- V**
  - velocity reinitialization (node) 81
  - volume force calculation (node) 213
- W**
  - wall (node)
    - particle tracing 67
  - wall correction (drag force), theory 230
  - websites, COMSOL 17

