Particle Tracing Module

User's Guide

Particle Tracing Module User's Guide

Contact Information

Visit the Contact COMSOL page at 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 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. Other useful links include:

- Support Center: www.comsol.com/support
- Product Download: www.comsol.com/product-download
- Product Updates: www.comsol.com/support/updates
- COMSOL Blog: www.comsol.com/blogs
- Discussion Forum: www.comsol.com/community
- Events: www.comsol.com/events
- COMSOL Video Gallery: www.comsol.com/video
- Support Knowledge Base: www.comsol.com/support/knowledgebase

Part number: CM022701

Contents

Chapter 1: Introduction

Chapter 2: Particle Tracing Modeling

Chapter 3: Mathematical Particle Tracing

Chapter 4: Charged Particle Tracing

Chapter 5: Particle Tracing for Fluid Flow

Chapter 6: Multiphysics Interfaces

Chapter 7: Glossary

1

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 forces. 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*:

- Bidirectionally Coupled Particle Tracing (study)
- Particle (Dataset)
- Particle Trajectories (plot) and Filter for Particle Trajectories
- Phase Portrait (plot)
- Poincaré Map (plot)
- Particle Evaluation (numerical)

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

	For an example of:
	• Particle Trajectories see *Ion Cyclotron Motion*: Application Library path: **Particle_Tracing_Module/Charged_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/Tutorials/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_rel(expr)` and `bndenv_rel(expr)` are analogous to `env(expr)` and `bndenv(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 STUDY TYPE
⚡ **AC/DC**				
⚛ **Particle Tracing**				
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
Charged Particle Tracing		cpt	3D, 2D, 2D axisymmetric	bidirectionally coupled particle tracing; time dependent

PHYSICS INTERFACE	ICON	TAG	SPACE DIMENSION	AVAILABLE STUDY TYPE
≋ **Fluid Flow**				
🔅 **Particle Tracing**				
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
△u **Mathematics**				
Mathematical Particle Tracing	🔅ₓ	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 2-4 for links to common sections and Table 2-5 to common feature nodes. You can also search for information: press F1 to open the **Help** window or Ctrl+F1 to open the **Documentation** window.

Where Do I Access the Documentation and Application Libraries?

A number of 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, 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 the features in the COMSOL Desktop. To learn more about a node in the Model Builder, or a window on the Desktop, click to highlight a node or window, then press F1 to open the Help window, which then displays information about that feature (or click a node in the Model Builder followed by the **Help** button (**?**). This is called *topic-based* (or *context) help*.

Win   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.

To open the **Help** window:

Mac
- In the **Model Builder** or **Physics Builder** click a node or window and then press F1.

Linux
- On the main toolbar, click the **Help** (?) button.
- From the main menu, select **Help>Help**.

Opening the Documentation Window

To open the **Documentation** window:

Win
- Press Ctrl+F1.
- From the **File** menu select **Help>Documentation** (▮).

To open the **Documentation** window:

Mac
- Press Ctrl+F1.

Linux
- On the main toolbar, click the **Documentation** (▮) button.
- From the main menu, select **Help>Documentation**.

THE APPLICATION LIBRARIES WINDOW

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

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

The Application Libraries Window in the *COMSOL Multiphysics Reference Manual*.

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

Win	• From the **Home** toolbar, **Windows** menu, click ( ) **Applications Libraries**. • From the **File** menu select **Application Libraries**. To include the latest versions of model examples, from the **File>Help** menu, select ( ) **Update COMSOL Application Library**.

Mac **Linux**	Select **Application Libraries** from the main **File>** or **Windows>** menus. To include the latest versions of model examples, from the **Help** menu select ( ) **Update COMSOL Application Library**.

CONTACTING COMSOL BY EMAIL

For general product information, contact COMSOL at info@comsol.com.

COMSOL ACCESS AND TECHNICAL SUPPORT

To receive technical support from COMSOL for the COMSOL products, please contact your local COMSOL representative or send your questions to support@comsol.com. An automatic notification and a case number are sent to you by email. You can also access technical support, software updates, license information, and other resources by registering for a COMSOL Access account.

COMSOL ONLINE RESOURCES

COMSOL website	www.comsol.com
Contact COMSOL	www.comsol.com/contact
COMSOL Access	www.comsol.com/access
Support Center	www.comsol.com/support
Product Download	www.comsol.com/product-download
Product Updates	www.comsol.com/support/updates
COMSOL Blog	www.comsol.com/blogs
Discussion Forum	www.comsol.com/community
Events	www.comsol.com/events
COMSOL Video Gallery	www.comsol.com/video
Support Knowledge Base	www.comsol.com/support/knowledgebase

Overview of the User's Guide

The *Particle Tracing Module User's Guide* gets you started with modeling using COMSOL Multiphysics with the Particle Tracing Module. 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 Fluid-Particle Interaction Interface, are described in this chapter.

2

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 nonrelativistic speeds.

	Electron Beam Divergence Due to Self Potential: Application Library path **Particle_Tracing_Module/Charged_Particle_Tracing/ electron_beam_divergence**

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. In such models, it is often best to start with a small number of particles, run the study, and then assess whether or not the effect is significant.

	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 Force and Dielectrophoretic Force.

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 Multiphysics, 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**

Flow type	Required features	
Sparse flow	Drag force	
Dilute flow (two-way coupling)	Drag force, fluid-particle interaction	
Dispersed flow (four-way coupling)	Drag force, fluid-particle interaction, particle-particle interaction	
Dense flow	Cannot be solved with the Particle Tracing Module	

Increasing particle mass and volume density
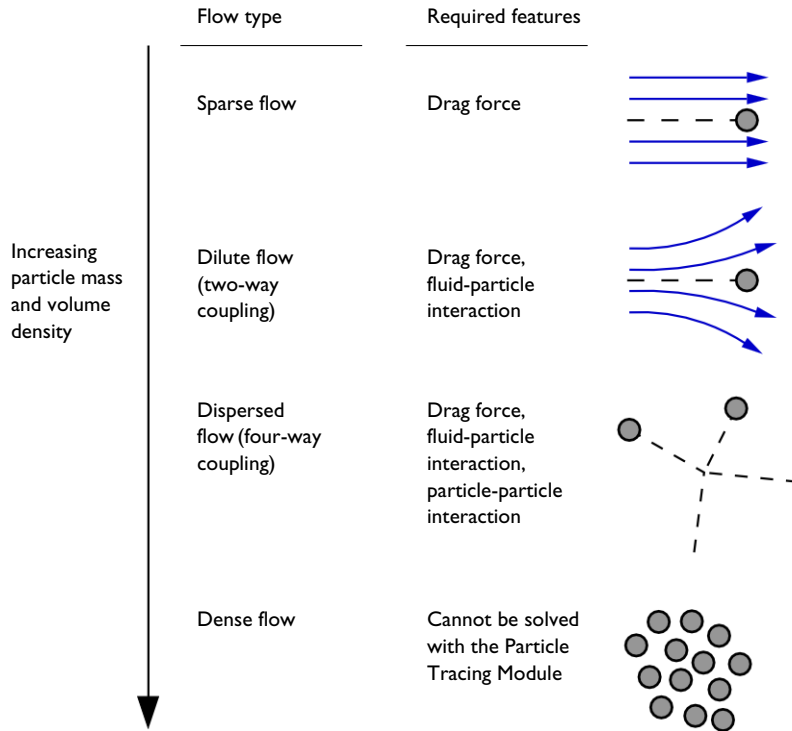
Figure 2-1: The physics features required to model sparse, dilute, and dispersed flows.

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 coupling 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 coupling 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. Hard-sphere collisions between model particles are not supported.

- 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.

COMPUTING PARTICLE TEMPERATURE AND MASS

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 **Additional Variables** section of the Settings window for the physics interface. 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.

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:

$$\text{Pe} = \frac{L|\mathbf{u}|}{D}$$

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: m^2/s) for spherical particles is, following Einstein's relation,

$$D = \frac{k_{\text{B}}T}{6\pi\mu r_{\text{p}}}$$

where

- μ (SI unit: Pa·s) is the fluid dynamic viscosity,
- $k_{\text{B}} = 1.380649 \times 10^{-23}$ 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 4×10^{-12} m^2/s. For a microfluidic device with characteristic size 1 mm and velocity of 1 m/s, this results in a Péclet number of 2.5×10^8. Handling such a large Péclet number with continuum methods is usually not practical (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. Molecular 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 Kn (dimensionless) is very small,

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

where λ (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.
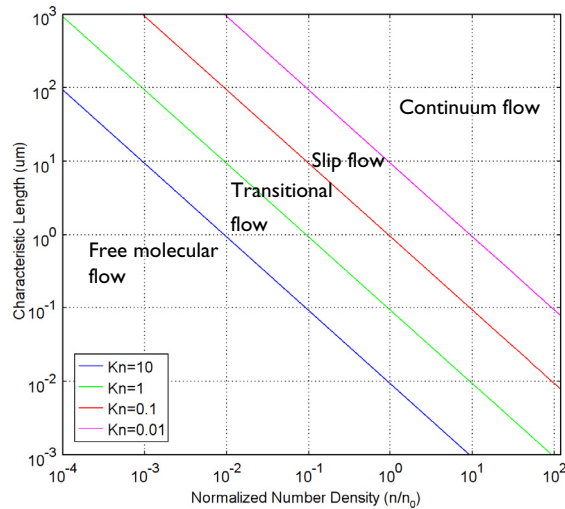
Figure 2-2: A plot showing the main fluid flow regimes for rarefied gas flows. Different regimes are separated by lines of constant Knudsen numbers. The number density of the gas is normalized to the number density of an ideal gas at a pressure of 1 atmosphere and a temperature of 0 °C (n0).

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 Couplings

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 nonrelativistic 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.

The Particle Field Interaction, Non-Relativistic interface is used in several Application Library examples:

- *Child's Law Benchmark*: Application Library path **Particle_Tracing_Module/Charged_Particle_Tracing/childs_law_benchmark**
- *Electron Beam Divergence Due to Self Potential*: Application Library path **Particle_Tracing_Module/Charged_Particle_Tracing/ electron_beam_divergence**
- *Thermionic Emission in a Planar Diode*: Application Library path **Particle_Tracing_Module/Charged_Particle_Tracing/planar_diode**

The Particle Field Interaction, Relativistic interface is used in the example *Relativistic Diverging Electron Beam*: Application Library path **Particle_Tracing_Module/Charged_Particle_Tracing/ electron_beam_divergence_relativistic**.

Modeling Tools

In this section:

> 🔍 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 second law of motion,

$$\frac{\mathrm{d}}{\mathrm{d}t}\left(m_{\mathrm{p}}\frac{\mathrm{d}\mathbf{q}}{\mathrm{d}t}\right) = \mathbf{F}$$

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,

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

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 the 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

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

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

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

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

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

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:

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

These are known as *Hamilton's equations.*

For example, for a free particle,

$$H = \sum_j p_j \dot{q}_j - \sum_j \frac{p_j^2}{2m_\mathrm{p}}$$

Substitution with Equation 2-2 and Equation 2-3 then yields

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

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

$$\frac{d\mathbf{q}}{dt} = \frac{1}{m_\mathrm{p}}\mathbf{p} \qquad \frac{d\mathbf{p}}{dt} = 0$$

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 2-3. 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.
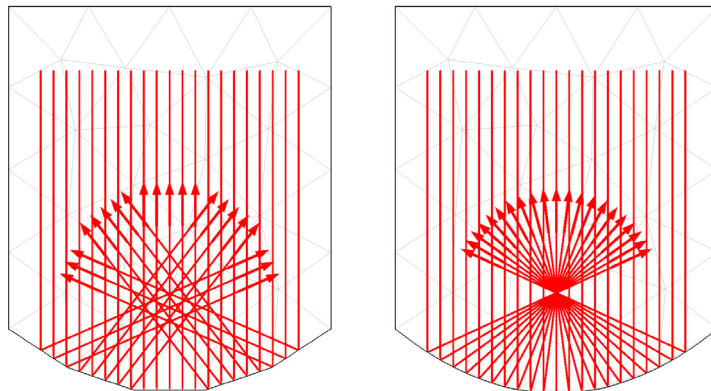
Figure 2-3: Comparison of particles being specularly reflected at a curved boundary represented using linear elements (left) and quadratic elements (right).

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 `.mphtxt` 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.

All of the variables described in this section are preceded by the physics interface identifier: usually `pt`, `cpt`, or `fpt`. For example, the particle index variable defined by The Mathematical Particle Tracing Interface would be called `pt.pidx`, whereas for The Particle Tracing for Fluid Flow Interface it would be called `fpt.pidx`. If multiple instances of the physics interface exist, the additional instances are followed by a number, e.g. `pt` and `pt2`.

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.

	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 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 st.

- The final status of the particle 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 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**, alpha. 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:

- **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**, `frel` (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>`.

Nonlocal Couplings

The purpose of a model is often to compute the sum, average, maximum value, or minimum value of a quantity over a group of 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.

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

  The treatment of `NaN` values in nonlocal minimum couplings can be platform-dependent, so use caution when evaluating the minimum over all particles including disappeared and unreleased 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 nonlocal couplings shown in Table 2-1.

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

NAME	DESCRIPTION
`pt.ptop1(expr)`	Sum over particles
`pt.ptop_all1(expr)`	Sum over all particles
`pt.ptaveop1(expr)`	Average over particles
`pt.ptaveop_all1(expr)`	Average over all particles
`pt.ptmaxop1(expr)`	Maximum over particles
`pt.ptmaxop_all1(expr)`	Maximum over all particles
`pt.ptminop1(expr)`	Minimum over particles
`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.ptminop1(expr, evalExpr)`	Evaluate at minimum over particles
`pt.ptminop_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.

- **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.

In the *Brownian Motion* tutorial, several different values of the **User defined** random number seed are used. The results show similar global quantities such as transmission probability, but the paths of individual particles are uncorrelated for each value of the random number seed.

Brownian Motion: Application Library path **Particle_Tracing_Module/ Tutorials/brownian_motion**

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 stochastic modeling of particle collisions with 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.

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 Couplings. 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.

▦	See *Electron Beam Divergence Due to Self Potential*, Application Library path **Particle_Tracing_Module/Charged_Particle_Tracing/ electron_beam_divergence**.

▦	If you have the AC/DC Module, also see *Relativistic Diverging Electron Beam*, Application Library path **Particle_Tracing_Module/ Charged_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

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

is solved, and so the variable **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:

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

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 or diameter (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.

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 from the following built-in distributions:

- **None**: enter a single value or expression.

- **Normal**, **Lognormal**, or **Uniform**: sample a number of values from the chosen distribution type. Enter the **Mean** (default 0) and **Standard deviation** (default 1).

  For example, 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)`.

- **List of values**: Type a list of numeric values directly. The valued do not need to be at regular intervals and do not need to be entered in any particular order.

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 Couplings 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.

Particle Tracing with Multiple Species

It is possible to solve for the trajectories of several different types of particle as part of the same analysis. For example:

- Use The Mathematical Particle Tracing Interface to model gravitational attraction between stars or planets of different sizes.

- Use The Charged Particle Tracing Interface to model the reactions between ions, electrons, and neutral atoms or molecules in a low-pressure environment. You can also model the Coulomb force between electrons and different species of ion.

- Use The Particle Tracing for Fluid Flow Interface to model separation of different types of biological cell. You can also model the filtration of sediment particles with different masses; the masses can have a number of discrete values or be sampled from a continuous distribution, such as a normal distribution.

You can define additional species in the model by adding more instances of the Particle Properties feature. One **Particle Properties** node is always present in the interface by default, and cannot be deleted. When you create additional **Particle Properties** nodes, they are always listed directly under the default **Particle Properties** node in the Model Builder.

If multiple types of particle are present in the same model, you can choose which species to release with every particle release feature, such as the Inlet, Release from Grid, and Secondary Emission features. Expand the **Released Particle Properties** section, which includes a list of all of the available **Particle Properties** nodes. Then select the node corresponding to the species that the feature should release.
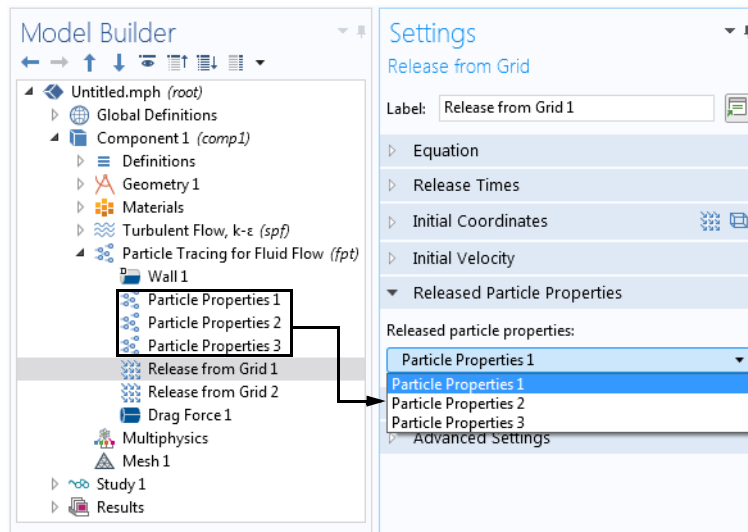
Figure 2-4: The Released Particle Properties section of the Settings window for Release from Grid.

EXAMPLE: COULOMBIC ATTRACTION BETWEEN IONS AND ELECTRONS

To model the Coulomb interaction between ions and electrons, use the Charged Particle Tracing interface and take the following steps:

1 In the **Model Builder** window, under the **Charged Particle Tracing (cpt)** interface click the default **Particle Properties** node.

2 Optionally, type Electrons in the **Label** text field to make the **Particle Properties** node more descriptive.

3 Note that the default value for the **Particle mass** m_p is me_const, a built-in physical constant for the electron mass. The default value for the **Charge number** Z is -1. Both of these values can be left as their defaults because they apply to electrons.

4 On the **Physics** toolbar, click **Global** and choose **Particle Properties**.

5 Optionally, type Protons in the **Label** text field.

6 For the **Particle mass** m_p enter mp_const, a built-in physical constant for the proton mass. For the **Charge number** Z enter 1.

7 On the **Physics** toolbar, click **Global** and choose **Release from Grid**.

8 Optionally, type Electron Release in the **Label** text field.

9 Enter in the initial position and velocity of the electron.

10 Click to expand the **Released Particle Properties** section. Note that the first species, **Electrons**, is selected by default.

11 On the **Physics** toolbar, click **Global** and choose **Release from Grid**.

12 Optionally, type Proton Release in the **Label** text field.

13 Enter in the initial position and velocity of the proton.

14 From the **Released particle properties** list, select **Protons**.

15 On the **Physics** toolbar, click **Domains** and choose **Particle-Particle Interaction**.

16 Select all domains. By default, **Coulomb** is selected from the **Interaction Force** list.

When the study is run, one electron and one proton will be released, and these two particles will be attracted to each other.

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**.

The **Particle type** can be set to render primary particles, secondary particles, both, or by a logical expression. The logical expression can be used to plot only particles of a certain species, like red blood cells or protons.

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.
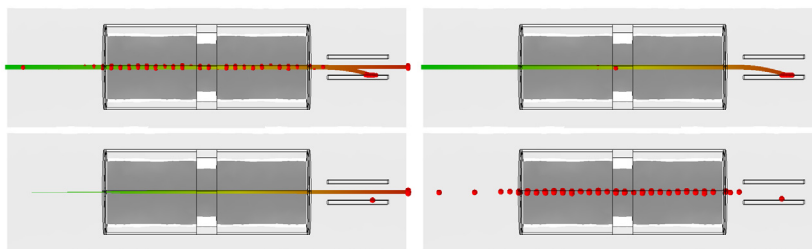
Figure 2-5: Neutralization of a H^+ beam by a rarefied buffer gas of Argon, showing all particles (top left), H^+ only (top right), H only (lower-left), and Ar^+ only (lower-right).

	If you have the Molecular Flow Module, see *Neutralization of a Proton Beam Through a Charge Exchange Cell*, Application Library path **Molecular_Flow_Module/Industrial_Applications_charge_exchange_cell**.

Particle Counters

A Particle Counter 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 .

Note that the solution at the stored times is equally valid in both cases; even though a time step is not 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.
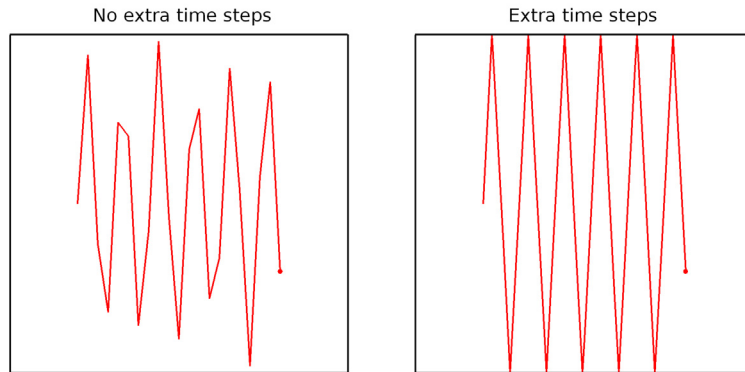
Figure 2-6: Comparison of the Particle Trajectories plot without extra time steps (left) and with extra time steps (right).

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.
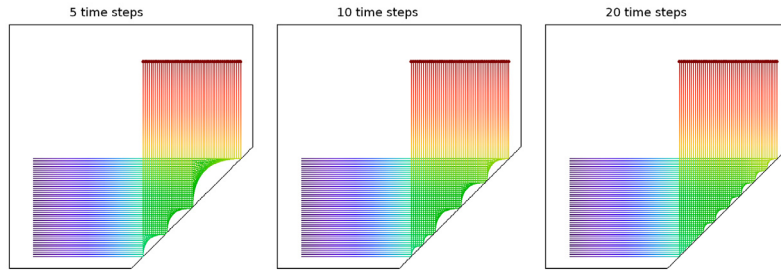
Figure 2-7: Comparison of the stored extra time steps for three different numbers of stored solution times.

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.

3

Mathematical Particle Tracing

This chapter describes the Mathematical Particle Tracing interface found under the **Mathematics** branch ($\triangle u$) 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 (△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.

In The Charged Particle Tracing Interface and The Particle Tracing for Fluid Flow Interface, only **Newtonian**, **Newtonian, first order**, and **Massless** are available. Most built-in forces require one of the Newtonian formulations.

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
nsr	Number of released secondary particles
nsrf	Number of released secondary particles at final time step
nsu	Number of unreleased secondary particles
nsuf	Number of unreleased secondary 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.

> [!] The **Store particle status data** check box should always be selected if **Automatic remeshing** or **Adaptive mesh refinement** is used in the study.

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 More Options** button ( ) on the **Model Builder** toolbar, and select **Advanced Physics Options** in the **Show More Options** dialog box).

Wall Accuracy Order

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.

Reuse Particle Degrees of Freedom

Select an option from the **Reuse particle degrees of freedom** list: **None** (the default), **All disappeared particles**, or **Disappeared secondary particles only**.

This setting controls how degrees of freedom are allocated for the emitted secondary particles in the model. If **None** is selected, then no particle degrees of freedom can be recycled; once any particle disappears, the corresponding dependent variables can no longer change.

If **All disappeared particles** is selected, then the degrees of freedom for any particle that has disappeared can be used later to release a new secondary particle. In this way, it is possible to set up a model in which a very large number of secondary particles are emitted over a long period of time, with the total number of secondary particles released exceeding the **Maximum number of secondary particles** specified in the **Particle Release and Propagation** section, as long as the rate of secondary particle emission does not exceed the rate of particle annihilation.

If **Disappeared secondary particles only** is selected, then only the preallocated secondary particle degrees of freedom can be recycled. Once a primary particle disappears, its degrees of freedom can no longer be used.

For example, consider a model in which 500 particles are released and the **Maximum number of secondary particles** is 100. If **None** is selected, then secondary particles can only be released in the model up to 100 times, no matter what happens to any of the particles in the model. If **All disappeared particles** is selected, then it is possible for more than 100 secondary particles to be active in the model at a later time, as long as some of the primary particles have disappeared in order to make their degrees of freedom available. If **Disappeared secondary particles only** is selected, then more than 100 secondary particles may be emitted during the course of the simulation, but no more than 100 secondary particles can exist at any one time.

⚠️ If the option **All disappeared particles** is selected, then some built-in static variables will not always give correct results. For example, the particle statistics for release time, stop time, and final status, which are generated by selecting the **Store particle status dat**a check box in the **Additional Variables** section, may give incorrect information at some solution times when the degrees of freedom are reused.

Arguments for Random Number Generation
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.

- 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.

Maximum Number of Wall Interactions per Time Step
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.

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**)
- Accumulator (for 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)
- Thermal Re-Emission
- Periodic Condition
- Accumulator (Boundary) (subnode to **Wall**, **Outlet**, and **Axial Symmetry**)
- Particle Counter

PAIRS

- Inlet
- Particle Continuity

- Release from Edge

- Release from Point

- Auxiliary Dependent Variable
- Particle Properties (default)
- Release from Grid
- Release from Data File

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

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**, **Isotropic 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.
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 θ is the angle between the direction of the reflected particle and the wall normal. The total particle kinetic energy is conserved.
Isotropic scattering	Select to bounce particles off a wall isotropically in random directions. 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 γ_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 General Reflection Settings.

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** γ (dimensionless) of the particle behaving according to the **Wall condition**. Otherwise the particle behaves according to the Otherwise setting.

📑	The value of γ should always be between 0 and 1.

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

- **Freeze** and γ 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 γ 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 $1\ \mu m$ 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` (or `pt.dp>1[um]`), 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_{new} in the field. So, to increment the value of psi by one when a particle touches or crosses a boundary, enter `psi+1` in the text field for psi_{new}.

ADVANCED SETTINGS

If the **Primary particle condition** is set to **Probability**, or if the **Diffuse scattering** or **Isotropic 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.

Q   About the Boundary Conditions for the Particle Tracing Interfaces

Thermal Re-Emission

The **Thermal Re-Emisison** feature diffusely reflects particles from a boundary, like the Wall with the **Diffuse scattering** wall condition. Unlike the **Diffuse scattering** condition, however, the **Thermal Re-Emission** feature also re-samples the particle speed from a distribution based on the temperature of the boundary.

Use this node to model the adsorption of molecules at a wall, and their subsequent re-emission into the modeling domain after reaching a state of thermal equilibrium with the surface.

WALL PROPERTIES

Enter the **Temperature** T (SI unit: K). If the temperature is computed by another physics interface then it can be selected from the list.

Enter a value or expression for the **Freezing probability** γ (dimensionless). The default is 0. This is the probability (a number from 0 to 1) that a particle will be absorbed by the wall and won't be re-emitted. For 0 all particles are reflected into the modeling domain, while for 1 all particles are absorbed.

⊕	About the Boundary Conditions for the Particle Tracing Interfaces

Periodic Condition

📄	The **Periodic Condition** node is only available when one of the following is selected as the **Formulation** in the physics interface **Particle Release and Propagation** section: **Newtonian**, **Newtonian, first order**, **Lagrangian**, or **Hamiltonian**.

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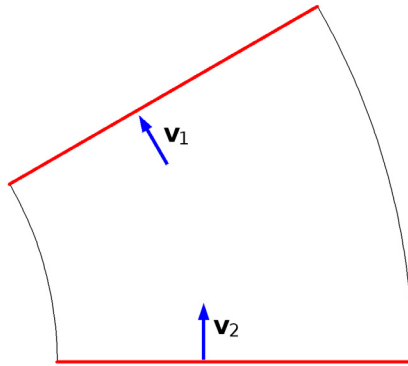
Figure 3-1: Transformation of particle velocity in a geometry with sector symmetry.

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.

In other words, if a particle were to get stuck in an infinite loop between the source and destination boundaries in a **Periodic Condition**, the particle would instead go through a large but finite number of wall interactions, then disappear if the interaction is still not resolved.

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).

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.

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.

Q | Accumulator Theory: Boundaries

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. Alternatively, select a particle release feature from the list, and then only the particles produced by that feature are counted.

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** γ (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 (t1, t2, 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).

Secondary Particle Speed

For **Secondary Emission** nodes added to boundary features, the **Secondary particle speed** list is shown if either **Isotropic hemisphere**, **Diffuse scattering**, or **Reflection of primary particle** is selected from the Initial velocity list. Select one of the following options from this list:

- For **Same as incident particle**, the initial speed of every released secondary particle is equal to that of the incident particle.

- For **Scaled by number of secondary particles** (the default), the initial speed of every released particle is equal to that of the incident particle, divided by the number or expression in the **Number of secondary particles** field.

- For **User defined**, enter a value or expression for the secondary particle speed V_s (SI unit: m). The default is 0.

For example, suppose that you set $N_s = 4$, select **Diffuse scattering** from the **Initial velocity** list, and then select **Scaled by number of secondary particles** from the **Secondary particle speed** list. If an incident particle with speed 2 m/s hits the wall, then 4 secondary particles will be released, with randomly generated velocity directions following the cosine law and initial speeds of 0.5 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** Δt (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** Δr (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.

RELEASED PARTICLE PROPERTIES

Select an option from the **Released particle properties** list. At least one instance of the Particle Properties node is always available, because it is a default feature. If other instances of the **Particle Properties** feature have been added to the model, then they can be selected from the list. Use this input to specify which type of secondary particle is released when modeling multiple particle species.

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 rp_0.

🔍	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, $9.10938356 \times 10^{-31}$ 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.

For example, the expression pt.mp*(pt.vx^2+pt.vy^2+pt.vz^2)/2 is the Lagrangian for a free particle in 3D, not subjected to any forces.

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. For **Massless**, 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 example, in 3D the Hamiltonian of a free particle is `(px^2+py^2+pz^2)/(2*pt.mp)`.

- 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 **Particle Properties** node is a default feature for all particle tracing physics interfaces. It is possible to add more than one instance of this **Particle Properties** node to the same model. When you do so, each instance corresponds to a different species of particle. For example, two **Particle Properties** nodes could represent ions and electrons, or white blood cells and bacteria.

When two or more species are in the same model, you can decide which species is released by each particle release feature, such as the Release, Inlet, or Release from Grid feature. To do so, select the appropriate **Particle Properties** node from the **Released particle properties** list in the physics feature **Released Particle Properties** section.

For more information, see Particle Tracing with Multiple Species in the Particle Tracing Modeling chapter.

Force

📄 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 F** (SI unit: N) based on space dimension.

For **Using susceptibility and field**:

- Select **Isotropic**, **Diagonal**, **Symmetric**, or **Full** and enter a value or expression in the field or matrix (based on space dimension) for the **Susceptibility** χ.

- Enter values or expressions for the **Field** Γ.

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. At least one Particle Properties node is always available, because it is a default feature. If other instances of the **Particle Properties** feature have been added to the model, then they can be selected from the list. Use this option to apply a force to a specific type or species 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^3. 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** Ω (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.

Q   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 and Accumulator (for Velocity Reinitialization) subnodes are 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 (for Velocity Reinitialization)

The **Accumulator** subnode is available from the context menu (right-click the Velocity Reinitialization parent node) or from the **Physics** toolbar, **Attributes** menu. Each **Accumulator** subnode defines a variable, called the accumulated variable, in the domains where the parent **Velocity Reinitialization** is applied. Whenever a particle is affected by the **Velocity Reinitialization**, whether its velocity is actually changed or not, the value of the accumulated variable in that element is incremented based on the value of the user-defined **Source** term R for the particle.

ACCUMULATOR SETTINGS

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

- For **Density** the accumulated variable is divided by the volume (in 3D) or area (in 2D) of the domain element in which the reinitialization condition is satisfied.

- For **Count** the source term is not divided by the mesh element volume or area.

Enter the **Accumulated variable name**. The default is rpv. 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 **Velocity Reinitialization** node in an instance of the Mathematical Particle Tracing interface using the default variable name rpv, the accumulated variable name might be pt.vre.vacc1.rpv.

Enter a **Source** R. The unit of the source term depends on the settings in the **Units** section. Whenever the velocity reinitialization condition for the parent node is

satisfied, the accumulated variable is incremented by the source term in the mesh element the particle occupies. If the **Accumulator type** is set to **Density**, the source term is divided by the volume of the element (in 3D) or area of the 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 mol/m^3.

 ⓠ   Accumulator Theory: Velocity Reinitialization

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 name is rpd. The accumulated variable is defined as <name>.<varname>, where <name> is the physics interface name and <varname> is the accumulated variable name. For example, in an instance of the 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^3.

🔍	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 make particles exert forces on 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** σ (SI unit: m), typically in the order of a few angstroms (the default value is 3.3×10^{-10} m or 3.3 Å).

- Enter the **Interaction strength** ε (SI unit: J), usually in the order of 10^{-21} J (the default value is 1.6×10^{-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 dest(qx), dest(qy), and 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 <name>.r, where <name> 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 ith 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 ith 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 of a particle that is exerting the force. For example, when computing the force on the ith particle, the expression `qx-dest(qx)` is the difference between the x-coordinate of some other particle (say, the jth particle) and the x-coordinate of the ith 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, 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
Select a **Distribution function**: **List of values** (default), **Uniform**, **Normal.**, or **Lognormal**.

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

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

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

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

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

	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 1dn 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 refine^dim, except for pyramids, where it is (4*refine^2-1)*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** ρ (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 underlying mesh.

	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 rerun 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.

> When releasing particles from domains with The Particle Tracing for Fluid Flow Interface, enter a value or expression for the **Velocity field u**. If another physics interface that computes the velocity field is present, then this interface can be selected from the list. This allows the particles to be released with initial velocity equal to that of the surrounding fluid.

- 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. The expression for the initial direction is automatically normalized to unity.

> 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_{vel} particles are released in a circle or sphere at each release point, each with the same speed. Enter the following:
  - **Speed** v (SI unit: m/s). The default is 1 m/s.
  - **Number of particles in velocity space** N_{vel} (dimensionless). The default is 200.

- For **Constant speed, hemispherical** a total of N_{vel} particles are released in a hemisphere or half circle at each release point, each with the same speed. Enter the:

  - **Speed** v (SI unit: m/s). The default is 1 m/s.

  - **Number of particles in velocity space** N_{vel} (dimensionless). The default is 200.

  - A direction vector for the **Hemisphere axis r** (dimensionless). This sets the direction of the north pole of the hemisphere or half circle.

- For **Constant speed, cone** a total of N_{vel} particles are released at each release point, each with the same speed. The velocity distribution is a cone of angle α for 3D components and an arc of half-angle α for 2D. Enter the:

  - **Speed** v (SI unit: m/s). The default is 1 m/s.

  - **Number of particles in velocity space** N_{vel} (dimensionless). The default is 200.

  - A direction vector for the **Cone axis r** (dimensionless). This sets the direction of the cone axis for 3D components or the centerline of the arc for 2D.

  - A **Cone angle** α (SI unit: rad).

- The **Constant speed, Lambertian** option is only available in 3D. It generates N_{vel} particles for each release point, each with the same speed. The velocity distribution is a hemisphere in which the angle distribution 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_{vel} (dimensionless). The default is 200.

  - A direction vector for the **Hemisphere axis 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 $N_v{}^{\text{sdim}}$, where `sdim` is the number of space dimensions.

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 pseudorandom 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[m/s]*(random(pt.pidx)+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.

RELEASED PARTICLE PROPERTIES

Select an option from the **Released particle properties** list. At least one instance of the Particle Properties node is always available, because it is a default feature. If other instances of the **Particle Properties** feature have been added to the model, then they can be selected from the list. Use this input to specify which type of particle is released when modeling multiple particle species.

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 **Specify particle mass** is selected from the **Particle size distribution** list 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 PARTICLE DIAMETER

This section is only available for The Particle Tracing for Fluid Flow Interface and is only shown when **Specify particle diameter** is selected from the **Particle size distribution** list in the physics interface **Additional Variables** section. The options in this section allow you to release many particles from each release point with different initial diameters.

The initial particle diameter 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 $d_{p,0}$. The default value is 1 μm.

Select **Normal** to create a 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 diameter** (default 1 μm) and **Particle diameter standard deviation** (default 0.1 μm).

Select **Lognormal** to create a log-normal distribution function. Then select one of the following options from the **Specify** list:

- For **Mean and standard deviation** (the default), enter the following:
  - **Number of values** N (dimensionless, default 1),
  - **Mean particle diameter** μ (SI unit: kg, default 1 μm), and
  - **Particle diameter standard deviation** σ (SI unit: m, default 0.1 μm).

- For **Count median diameter and geometric standard deviation**, enter the followong:
  - **Number of values** N (dimensionless integer, default 1),
  - **Count median diameter** CMD (SI unit: m, default 1 μm), and
  - **Geometric standard deviation** GSD (dimensionless, default 1.1).

- For **Sauter mean diameter and geometric standard deviation**, enter the following:

  - **Number of values** N (dimensionless integer, default 1),

  - **Sauter mean diameter** CMD (SI unit: m, default 1 μm), and

  - **Geometric standard deviation** GSD (dimensionless; default 1.1).

Alternatively, select **List of values** to enter a list of numerical values for the initial particle diameter directly.

<table>
<tr><td>🔍</td><td>Types of Particle Size Distribution in Theory for the Particle Tracing for Fluid Flow Interface.</td></tr>
</table>

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** $n_{\mathrm{n},0}$ (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 the **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. If the check box is selected, choose an option from the Background velocity check box: **From rotating frame** (the default) or **User defined**. If **From rotating frame** is selected then there should be an active instance of the Rotating Frame feature in the model. If **User defined** is selected, enter values or expressions for the components of the background velocity $\mathbf{v}_b$ (SI unit: m/s) directly. Thus, if this check box is cleared, particle velocity is always initialized with respect to the moving frame, which may be noninertial. If the check box is selected, the velocity is initialized with respect to the laboratory frame, i.e. a fixed noninertial 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**, **Released Particle 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 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.
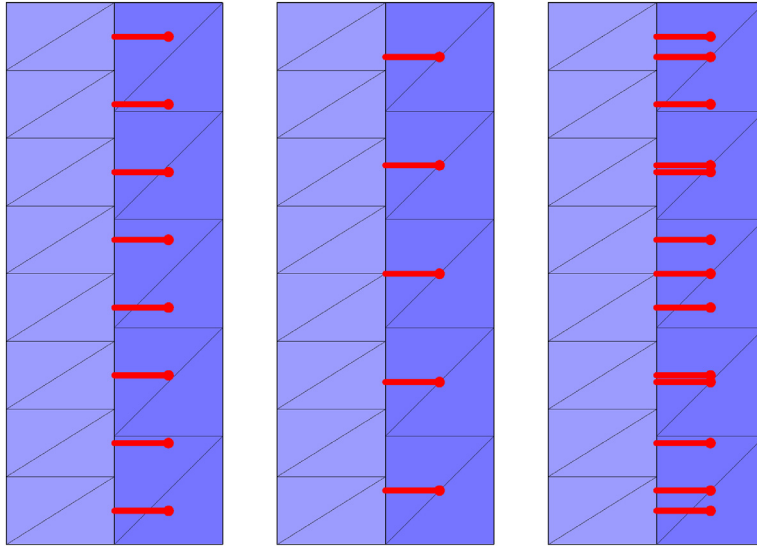
Figure 3-2: Mesh-based release of particles from a source boundary (left), destination boundary (middle), or both source and destination (right). In each rectangle, the source boundary is on the side of the lighter-colored mesh.

INITIAL VELOCITY

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

- 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 u**. If another physics interface that computes the velocity field is present, then this interface 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_{vel} (dimensionless), and the **Hemisphere axis 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_{vel} (dimensionless), the **Cone axis r**, and the **Cone angle** α (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_{vel} (dimensionless), and the **Hemisphere axis 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.

- For **Thermal**, enter a value or expression for the **Temperature** T of the boundary (SI unit: K). The default is 293.15 K. If the temperature is computed by another physics interface in the model then it can be selected from the list. The **Thermal** option samples the initial particle velocity from a distribution in the same manner as the Thermal Re-Emission boundary condition. For more details see About the Boundary Conditions for the Particle Tracing Interfaces in the Theory for the Mathematical Particle Tracing Interface section.

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, nx, ny, and nz) 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 pseudorandom 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[m/s]*(random(pt.pidx)+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.

ⓘ	About the Inlet Boundary Conditions

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 rpi.

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^3.

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.

The **Outlet** is essentially a simplified Wall with that can only absorb particles, not reflect them.

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** choose 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**.

U N I T S

Select a **Dependent variable quantity** from the list; the default is **Dimensionless [I]**. 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.

Except for the **Initial Position** section described below, all settings are the same as for the Release node.

I N I T I A L   P O S I T I O N

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.

Except for the **Initial Position** section described below, all settings are the same as for the Release node.

Release from Grid

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

Except for the **Initial Position** section described below, all settings are the same as for the Release node.

INITIAL COORDINATES

Select an option from the **Grid type** list: **All combinations** (the default), **Specified combinations**, **Cylindrical** (3D only), or **Hexapolar** (3D only).

Linear, Rectangular, and Arbitrary Grids

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

If **Specified combinations** is selected, the number of initial coordinates entered for each space dimension must be equal, and the total number of 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:

- $q_{x,0}$ = range(0,1,3)
- $q_{y,0}$ = 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.

Cylindrical Grids

For **Cylindrical**, enter coordinates for the **Center location** $\mathbf{q}_c$ (SI unit: m). By default, the distribution is centered at the origin. Then enter the components of the **Cylinder axis direction** $\mathbf{r}_c$ (dimensionless). The particles will be released at specified radial distances and angles in the plane containing the point $\mathbf{q}_c$ and orthogonal to the direction $\mathbf{r}_c$.

Select an option from the **Radial distribution** list: **Uniform radius intervals** (the default), **Uniform number density**, or **User defined**.

For **Uniform radius intervals** or **Uniform number density** enter a value or expression for the **Radius** R_c (SI unit: m). The default is 1 m. Then enter a positive integer for the **Number of radial positions** N_c (dimensionless). The default is 5.

For **User defined** enter a list of **Radial coordinates** q_r (SI unit: m) directly. An arbitrary number of radial coordinates can be entered in the list. The default is 1 m.

The effect of the **Radial distribution** setting on the resulting particle placement is illustrated in Figure 3-3. For **Uniform radius intervals**, the distances between the concentric rings of particles are all equal, but the number density of particles is greater at the center of the distribution than at the outer edge. The option **Uniform number density** corrects this imbalance by defining nonuniform increments in the radial position between the concentric rings.

Enter a positive integer for the **Number of angles** N_ϕ (dimensionless). The default is 10.

The total number of particles released (before accounting for distributions of particle velocity or auxiliary dependent variables at each release point) is $N_c \times N_\phi + 1$ because a single particle is also released at the center of the distribution.

Figure 3-3: Comparison of the cylindrical release in which particles are positioned at uniform radius intervals (left), with uniform number density (center), or at user-defined radii (right).

Hexapolar Grids

For **Hexapolar**, enter coordinates for the **Center location** $\mathbf{q}_c$ (SI unit: m). By default, the distribution is centered at the origin. Then enter the components of the **Cylinder axis direction** $\mathbf{r}_c$ (dimensionless). The particles will be released at specified radial distances and angles in the plane containing the point $\mathbf{q}_c$ and orthogonal to the direction $\mathbf{r}_c$.

Enter a value or expression for the **Radius** R_c (SI unit: m). The default is 1 m. Then enter a positive integer for the **Number of radial positions** N_c (dimensionless). The default is 5.

Unlike the **Cylindrical** grid types described in the previous section, each concentric ring of the **Hexapolar** grid contains a different number of particles, as shown in Figure 3-4. One particle is always released at the center. The first ring of particles surrounding the center has **6** particles arranged in a regular hexagon. Each ring of particles beyond the first has **6** more particles than the ring preceding it, with the particles arranged in a regular polygon. The radius increments between consecutive rings are uniform.

The total number of grid points generated is

$$N_{\text{grid}} = 1 + 3N_c(N_c + 1)$$

Thus the relationship between the number of radius intervals and the total number of grid points is quadratic.
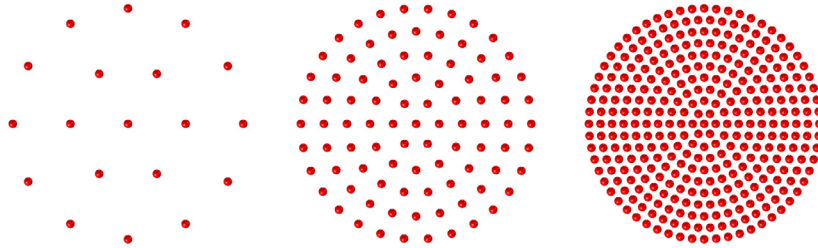
Figure 3-4: Comparison of hexapolar grids with 2 rings (left), 5 rings (middle), and 10 rings (right).

Previewing Grid Points

In the **Initial Coordinates** section, you can click the **Preview Initial Coordinates** and **Preview Initial Extents** buttons to visualize the initial particle positions. Clicking **Preview Initial Coordinates** will cause a point to appear in the Graphics window for every release position. Clicking **Preview Initial Extents** will cause a bounding box to appear, indicating the spatial extents of the released particles. Examples are shown in Figure 3-5 and Figure 3-6.
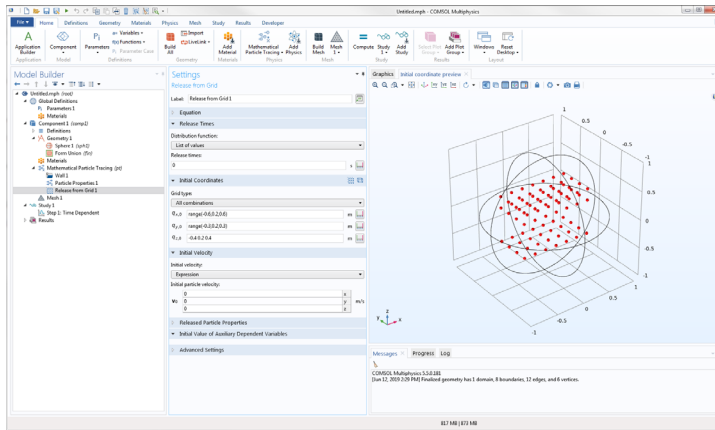
Figure 3-5: Graphics window after clicking the Preview Initial Coordinates button.
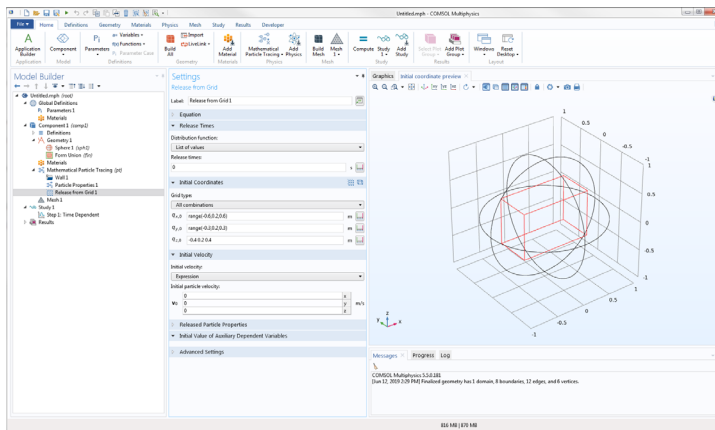
Figure 3-6: Graphics window after clicking the Preview Initial Extents button.

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**, **Released Particle 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 **Specify particle mass** is selected from the **Particle size distribution** list 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 PARTICLE DIAMETER

This section is only available for The Particle Tracing for Fluid Flow Interface and is only shown when **Specify particle diameter** is selected from the **Particle size distribution** list in the physics interface **Additional Variables** section.

The initial particle diameter 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

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:

$$\frac{\mathrm{d}}{\mathrm{d}t}(m_\mathrm{p}\mathbf{v}) = \mathbf{F}$$

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 total force exerted on the particle (SI unit: N). Each force is either specified directly or using a susceptibility multiplied by a suitable field:

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

where χ is the susceptibility tensor and Γ 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 relativistic 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:

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

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:

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

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 the coupled equations

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

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:

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

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 position 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 (noninertial) 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^2) that is rotating at angular velocity Ω (SI unit: rad/s) about a center of rotation with position $\mathbf{r}_{bp}$ (SI unit: m). Note that Ω is a vector quantity that also indicates the orientation of the axis of rotation and the sense of rotation (clockwise or counterclockwise). The total fictitious force $\mathbf{F}_{fr}$ (SI unit: N) exerted on a particle in this noninertial frame of reference is (Ref. 1)

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

TABLE 3-3: FICTITIOUS FORCES IN NONINERTIAL FRAMES

FORCE	DEFINITION	COMMENTS
Centrifugal force	$\mathbf{F}_{cen} = m_p\Omega \times (\mathbf{r} \times \Omega)$	Present in all rotating frames
Coriolis force	$\mathbf{F}_{cor} = 2m_p\mathbf{v} \times \Omega$	From particle motion in a rotating frame
Euler force	$\mathbf{F}_{eul} = m_p\mathbf{r} \times \dot{\Omega}$	From angular acceleration of the rotating frame
Linear acceleration force	$\mathbf{F}_{acc} = -m_p\mathbf{W}$	Not currently supported

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

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

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 $\mathbf{F}_{\mathrm{acc}}$ 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 $\mathbf{v}_{i,\mathrm{in}}$ and in the rotating frame $\mathbf{v}_{i,\mathrm{rot}}$ is

$$\mathbf{v}_{\mathrm{i,\,in}} \ = \ \mathbf{v}_{\mathrm{i,\,rot}} - \mathbf{v}_{\mathrm{f}}$$

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

$$\mathbf{v}_{\mathrm{f}} \ = \ \Omega \times \mathbf{r}$$

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:

$$\frac{\mathrm{d}\mathbf{q}}{\mathrm{d}t} \ = \ \mathbf{v}_0$$

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:

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

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 ith direction,
- m_p (SI unit: kg) is the particle mass,
- T_0 (SI unit: K) is the temperature, and
- $k_B = 1.380649 \times 10^{-23}$ J/K is the Boltzmann constant.

So the total distribution function is:

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

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

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

This can clearly generate a large number of particles because the total number of initial velocities are N_v^{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_{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:

$$v_x = c\cos\theta$$
$$v_y = c\sin\theta$$

where θ goes from 0 to 2π in N_{vel} steps.

In 3D the velocity distribution is given by:

$$v_x = c\sin\theta\cos\varphi$$
$$v_y = c\sin\theta\sin\varphi$$
$$v_z = c\cos\theta$$

The azimuthal angle φ is uniformly distributed from 0 to 2π. The polar angle θ is sampled from the interval $[0, \pi]$ with probability density proportional to $\sin\theta$. 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 θ goes from 0 to π and in 3D θ goes from 0 to $\pi/2$. The angle θ 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 θ goes from 0 to α. The angle 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 θ has a probability density proportional to $\sin\theta$; for the

Lambertian distribution the probability density is instead proportional to $\sin \theta \cos \theta$. 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:

$$\mathbf{F}_i = \frac{e^2}{4\pi\varepsilon_0} \sum_{j=1}^{N} Z_i Z_j \frac{(\mathbf{r}_i - \mathbf{r}_j)}{|\mathbf{r}_i - \mathbf{r}_j|^3}$$

where

- $e = 1.602176634 \times 10^{-19}$ C is the elementary charge,
- $\varepsilon_0 = 8.854187817 \times 10^{-12}$ F/m is the permittivity of vacuum,
- $\mathbf{r}_i$ is the position vector of the ith particle (SI unit: m),
- $\mathbf{r}_j$ is the position vector of the jth 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:

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

where r is the distance between the particles (SI unit: m), ε is the interaction strength (SI unit: Nm), and σ is the collision diameter (SI unit: m).
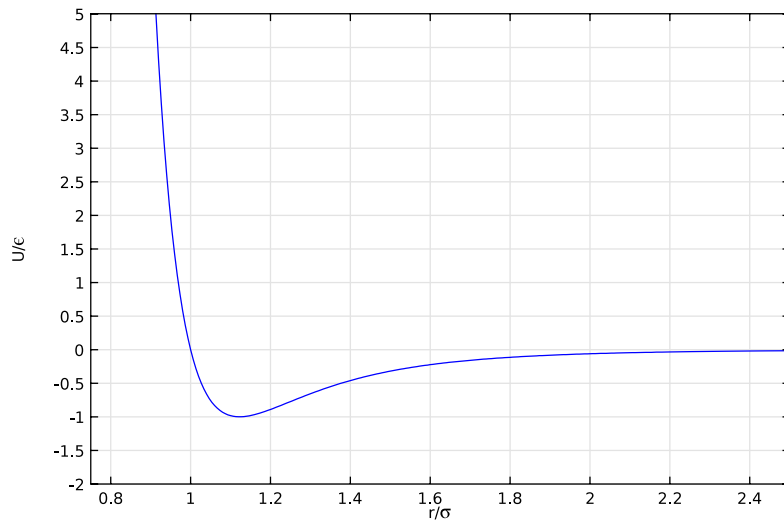
Figure 3-7: Plot of the Lennard-Jones potential normalized to the interaction strength versus the nondimensional radial distance.
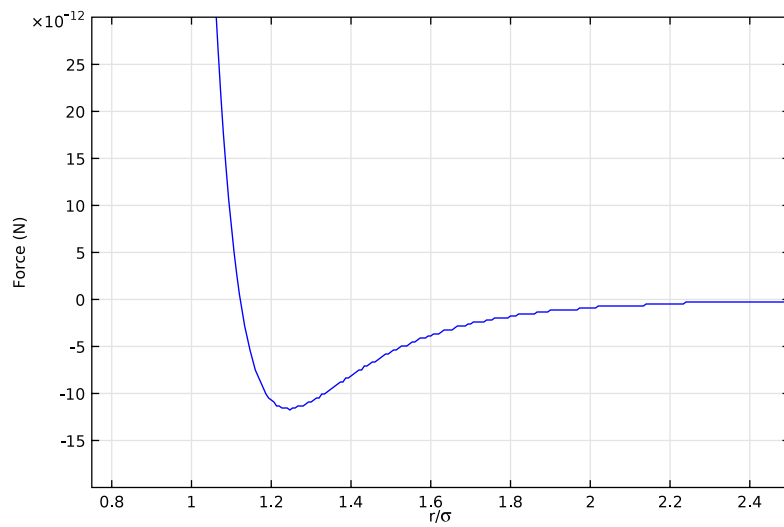
Figure 3-8: Plot of the Lennard-Jones force versus nondimensionalized radial distance between particles.

Using the fact that the force is computed as:

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

the expression for the force on the ith particle becomes:

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

A typical interaction strength is around $1.6 \cdot 10^{-21}$ J and a typical collision diameter is around $3.3 \cdot 10^{-10}$ 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 3-8.

LINEAR ELASTIC FORCE

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

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

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-defined 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 and Isotropic Scattering

The particle is reflected from the wall with a pseudorandomly generated velocity vector. Specifically, for the particle position:

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

and for the particle velocity in 3D:

$$v_{t1} = |\mathbf{v}_c| \sin\theta \cos\varphi$$
$$v_{t2} = |\mathbf{v}_c| \sin\theta \sin\varphi$$
$$v_n = |\mathbf{v}_c| \cos\theta$$

where θ is the polar angle and φ is the azimuthal angle. For both diffuse and isotropic scattering, the azimuthal angle is uniformly distributed in the interval $[0, 2\pi]$.

For diffuse scattering, the probability distribution function of the polar angle follows the cosine law, which states (Ref. 2) that the flux dn of reflected particles within a differential solid angle $d\omega$ is proportional to the cosine of the polar angle,

$$dn \propto \cos\theta d\omega$$

and noting that the differential solid angle is

$$d\omega = \sin\theta d\theta d\varphi$$

the probability distribution function of azimuthal and polar angles within a hemisphere about the surface normal $\mathbf{n}$ is

$$f(\theta, \varphi) = \frac{1}{\pi} \sin\theta \cos\theta$$

A value of the polar angle can be sampled from this distribution function by first defining a uniform random number Γ between 0 and 1. Then the polar angle is

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

For isotropic scattering, the flux of reflected particles is the same for any differential solid angle $d\omega$; the flux is not proportional to $\cos\theta$. The probability distribution function is

$$f(\theta, \varphi) = \frac{1}{2\pi}\sin\theta$$

Again defining a uniform random number Γ between 0 and 1, the polar angle is now

$$\theta = \text{acos}(\Gamma)$$

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

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

where for diffuse scattering the angle θ is defined as:

$$\theta = \text{asin}(\Gamma) \qquad \Gamma \in [-1, 1]$$

or for isotropic scattering,

$$\theta = \Gamma \qquad \Gamma \in \left[-\frac{\pi}{2}, \frac{\pi}{2}\right]$$

where Γ is uniformly distributed within the given interval.

Mixed Diffuse and Specular Reflection
The particle has probability γ 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.

THERMAL RE-EMISSION

At a boundary with the Thermal Re-Emission feature, particles are reflected into the modeling domain as if they were adsorbed at the wall and re-emitted with the wall temperature.

In 2D, the reflected particle speed is sampled from a distribution given by

$$f(V) = \sqrt{\frac{2}{\pi}} \left(\frac{m_p}{2k_B T} \right)^{3/2} V^2 \exp\left(-\frac{m_p V^2}{2k_B T} \right)$$

In 3D, the speed distribution is

$$f(V) = \left(\frac{m_p}{k_B T} \right)^2 \frac{V^3}{2} \exp\left(-\frac{m_p V^2}{2k_B T} \right)$$

where

- V (SI unit: m/s) is the speed of the reflected particle,
- T (SI unit: K) is the wall temperature,
- m_p (SI unit: kg) is the particle mass, and
- $k_B = 1.380649 \times 10^{-23}$ J/K is the Boltzmann constant.

The distribution of reflected particle directions follows Knudsen's cosine law; it is the same direction distribution as the **Diffuse scattering** condition for the Wall feature.

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** text 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

$$\mathrm{rpd_{count}} = \mathrm{rpd_{density}} \times V$$

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:

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

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:

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

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** text 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

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

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:

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

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 3-4: 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 nonlocal couplings on the selection of the parent physics feature, such as the **Wall** feature to which the **Accumulator** is added. The following expressions for the global variables are used.

TABLE 3-5: 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`.

Accumulator Theory: Velocity Reinitialization

The Accumulator (for Velocity Reinitialization) feature transfers information from particles to the domain they occupy. Each **Accumulator** defines a variable, called the accumulated variable, on the domains in the selection of the parent Velocity Reinitialization node. 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** text field in the **Accumulator Settings** section of the settings window. The default variable name is rpv.

The options in the **Accumulator type** list are **Density** and **Count**. If **Count** is selected, then whenever the parent feature's velocity reinitialization condition is satisfied, the accumulated variable in the element occupied by the particle is incremented by the value of the source term,

$$\text{rpv}_{\text{new}} \; = \; \text{rpv} + R$$

If instead **Density** is selected, the source term is first divided by the volume (in 3D) or area (in 2D) of the mesh element the particle is in,

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

It is not necessary for the particle velocity to change for the accumulation to occur; the reinitialization condition merely needs to be satisfied. For example, accumulation can occur even if **None** is selected from the **Effect on primary particle** list in the parent **Velocity Reinitialization** node.

A simple demonstration of the **Accumulator** subnode for the **Velocity Reinitialization** node is illustrated in Figure 3-9 below. A single particle is released into a square domain with initial velocity $(1,0.5)$ m/s. From left to right, the particle trajectory is shown at solution times $t = 0.25$ s, $t = 0.50$ s, $t = 0.75$ s, and $t = 1.00$ s. The accumulated variable is shown as a surface plot and the mesh element boundaries are also shown. The reinitialization criterion is qx>0.6&&pt.vx>0. It is clear that the accumulation is applied uniformly over the mesh element the particle occupies at the time step when the reinitialization condition is met, at about $t = 0.60$ s.
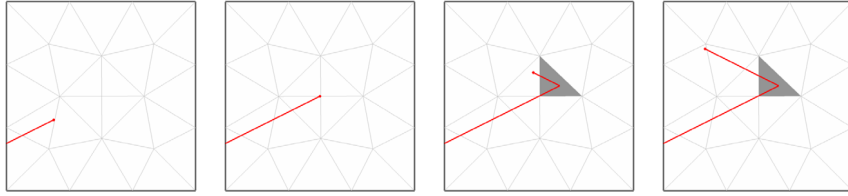
Figure 3-9: Velocity reinitialization of a single particle with an accumulator.

References for the Mathematical Particle Tracing Interface

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

2. John Greenwood, "The correct and incorrect generation of a cosine distribution of scattered particles for Monte-Carlo modeling of vacuum systems," *Vacuum*, vol. 67, no. 2, pp. 217-222, 2002.

4

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**. This setting affects the way particle-field interactions are modeled, and also affects the values of certain built-in accumulated variables.

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, except as follows.

Maximum Number of Consecutive Null Collisions

The **Maximum number of consecutive null collisions** (dimensionless) is a positive integer. The default is 100. This value is used to cap the number of collisions that a single particle can undergo within a single time step taken by the solver when performing a Monte Carlo collision simulation. This value is only used in the model if the Collisions node is present and **Null collision method, cold gas approximation** is selected from the **Collision detection** list.

For more information on the mathematics involved, see The Null Collision Method in Theory for the Charged 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

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 2-4 for links to common sections and Table 2-5 to common feature nodes. You can also search for information: press F1 to open the **Help** window or Ctrl+F1 to open the **Documentation** window.

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

- Accumulator (Boundary)
- Accumulator (Domain)
- Accumulator (for Velocity Reinitialization)
- Auxiliary Dependent Variable
- Force
- Inlet
- Nonlocal Accumulator
- Outlet
- 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
- Thermal Re-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.

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.

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

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 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.

Select an option from the **Collision detection** list: **At time steps taken by solver** (the default) or **Null collision method, cold gas approximation**.

	The two collision detection algorithms available from the **Collision detection** list are vastly different.
	If **At time steps taken by solver** is selected, collisions can only be detected at the beginning of a time step taken by the time-dependent solver, and the collision frequency is assumed to be constant over each step. Usually it is necessary to specify a maximum time step size or fixed step size in the solver settings, to ensure the steps taken by the solver are small relative to the free time between collisions.
	If **Null collision method, cold gas approximation** is selected, collisions can be detected at random times between the time steps taken by the solver. This option allows more accurate collision modeling when the steps taken by the solver are comparable to or larger than the free time between collisions. The tradeoff is that the thermal velocity distribution in the background gas is neglected, so this option is only applicable to highly energetic model particles.

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** ν (SI unit: Hz). The default is 2×10^6 Hz.

- For **Cross section** enter a value or expression for the **Collision cross section** σ (SI unit: m^2). The default is 3×10^{-19} m^2.

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: $1/m^3$). The default is 10^{20} $1/m^3$. 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_{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_{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.

See Elastic for information about the **Collision Statistics** and **New Value of Auxiliary Dependent Variables** sections.

COLLISION FREQUENCY

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

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.

See Elastic for information about the **Collision Statistics** and **New Value of Auxiliary Dependent Variables** sections.

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, which can be any Particle Properties node that has been added to the physics interface. This controls the type of species that is introduced via secondary particle emission during the attachment reaction.

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.

See Elastic for information about the **Collision Statistics** and **New Value of Auxiliary Dependent Variables** sections.

COLLISION FREQUENCY

These settings are the same as for the Elastic subnode with additional options to enter a value for the **Energy loss** ΔE (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 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, which can be any Particle Properties node that has been added to the physics interface.

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.

See Elastic for information about the **Collision Statistics** and **New Value of Auxiliary Dependent Variables** sections.

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** χ (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 node that has been added to the physics interface.

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.

See Elastic for information about the **Collision Statistics** and **New Value of Auxiliary Dependent Variables** sections.

COLLISION FREQUENCY

These settings are the same as for the Elastic subnode with additional options to enter a value for the **Energy loss** ΔE (SI unit: J), enter a value for the **Scattering angle in the center of mass system** χ (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 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 node that has been added to the physics interface.

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.

See Elastic for information about the **Collision Statistics** and **New Value of Auxiliary Dependent Variables** sections.

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.

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** γ (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** Δt (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** Δr (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.

RELEASED PARTICLE PROPERTIES

This section is available if the **Include secondary emission** check box is selected. Select an option from the **Inherit Properties** list. At least one Particle Properties node is always available, because it is a default feature. If other instances of the **Particle Properties** feature have been added to the model, then they can be selected from the list. Use this setting to determine what set of properties is given to the secondary particles that are released by this feature.

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** σ (SI unit: m^2). The default is 3×10^{-19} m^2.
- **Background number density** N_d (SI unit: $1/m^3$). The default is 10^{20} $1/m^3$.

If **Collision frequency** is selected, enter the **Collision frequency** ν (SI unit: $1/s$). The default value is 2×10^6 Hz.

FLUID PROPERTIES

Enter coordinates for the **Velocity field u** (SI unit: 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** ρ (SI unit: kg/m^2) 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: m^4/s^2). The default value is 1 MeV·cm^2/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** χ_c (SI unit: rad). The default is $0.1°$. 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 `Nc` preceded by the scope of the **Nuclear Stopping** node, for example `cpt.pmi1.ns1.Nc`.

Ⓠ | Particle-Matter Interaction Theory

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**, **Released Particle 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, that is, 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 yields 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 $1 \, mA/m^2$. 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 ($\varepsilon_{rms,1}$, β_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** $\mathbf{t}_1$ (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** β (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 β causes the particle distribution in transverse position space to be broadened and the distribution in transverse velocity space to be tightened.

• The **1-RMS beam emittance** ε_{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 mA/m^2. For symmetric beams the brightness can 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** α (dimensionless). The default is 0. When this value is nonzero, 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 γ (SI unit: $1/m$) is not specified because it is derived from α and β.

- 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** θ (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
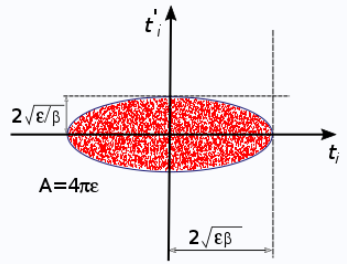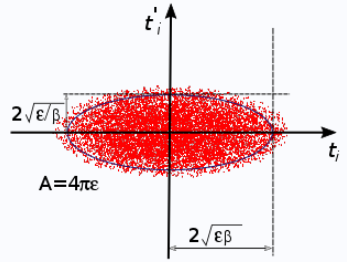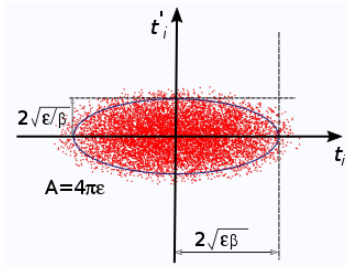
SAMPLING	ORIENTATION	TRANSVERSE VELOCITY SPECIFICATION	IMAGE	AVAILABLE INPUT
KV	Upright	Twiss		β, ε_{rms} β, B
Waterbag	Upright	Twiss		β, ε_{rms} β, B
Parabolic	Upright	Twiss		β, ε_{rms} β, B

TABLE 4-1: SUMMARY OF INITIAL TRANSVERSE VELOCITY OPTIONS
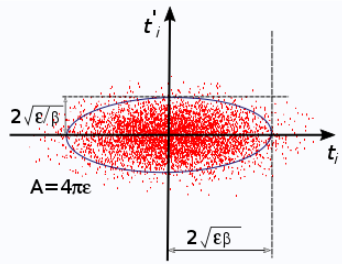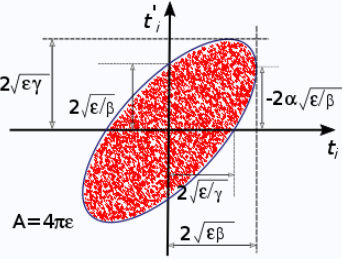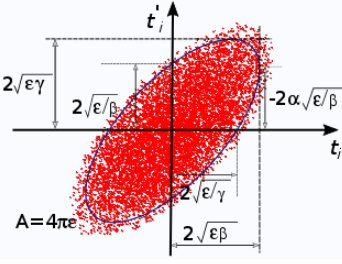
SAMPLING	ORIENTATION	TRANSVERSE VELOCITY SPECIFICATION	IMAGE	AVAILABLE INPUT
Gaussian	Upright	Twiss		β, ε_{rms} β, B
KV	Not upright	Twiss		β, ε_{rms}, α β, B, α
Waterbag	Not upright	Twiss		β, ε_{rms}, α β, B, α

TABLE 4-1: SUMMARY OF INITIAL TRANSVERSE VELOCITY OPTIONS

SAMPLING	ORIENTATION	TRANSVERSE VELOCITY SPECIFICATION	IMAGE	AVAILABLE INPUT
Parabolic	Not upright	Twiss		β, ε_{rms}, α β, B, α
Gaussian	Not upright	Twiss		β, ε_{rms}, α β, B, α
KV	Upright	Dimensions		x_m, x_m'

TABLE 4-1: SUMMARY OF INITIAL TRANSVERSE VELOCITY OPTIONS
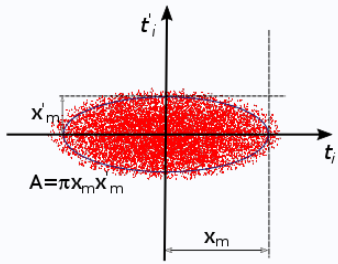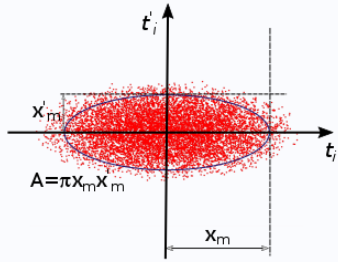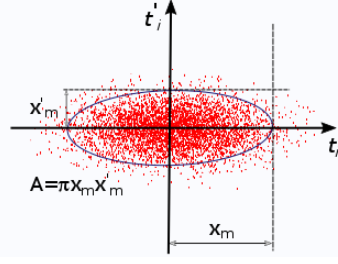
SAMPLING	ORIENTATION	TRANSVERSE VELOCITY SPECIFICATION	IMAGE	AVAILABLE INPUT
Waterbag	Upright	Dimensions		x_m, x_m'
Parabolic	Upright	Dimensions		x_m, x_m'
Gaussian	Upright	Dimensions		x_m, x_m'

TABLE 4-1: SUMMARY OF INITIAL TRANSVERSE VELOCITY OPTIONS
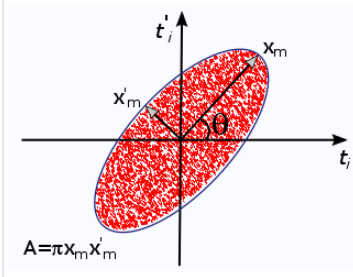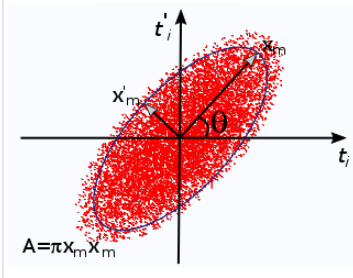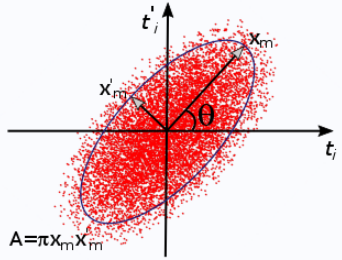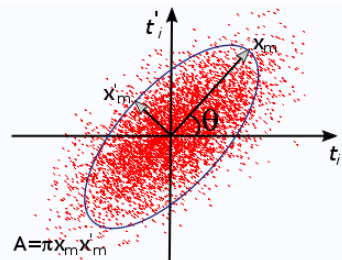
SAMPLING	ORIENTATION	TRANSVERSE VELOCITY SPECIFICATION	IMAGE	AVAILABLE INPUT
KV	Not upright	Dimensions		x_m, x_m', θ
Waterbag	Not upright	Dimensions		x_m, x_m', θ

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', θ
Gaussian	Not upright	Dimensions		x_m, x_m', θ

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** σ (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** σ (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: **Released Particle 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** Φ (SI unit: V). The default expression is 4.5 V.

Enter a value or expression for the **Effective Richardson constant** A^* (SI unit: $A/(m^2 \cdot K^2)$). The default expression is 110 $A/(m^2 \cdot K^2)$.

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 interior 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

▥	*Thermionic Emission in a 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 SPECIES

This section is shown when **Newtonian** or **Newtonian, first order** is selected as the **Formulation**. Choose an option from the **Particle species** list. The default is **User defined**, allowing the mass and charge number to be specified directly. The remaining options are detailed in Table 4-2. The values in the **Mass** column show the numeric values of the particle mass, as well as any built-in physical quantities for the masses of these species.

TABLE 4-2: BUILT-IN SPECIES FOR CHARGED PARTICLE TRACING

NAME	MASS (KG)	CHARGE NUMBER
Electron	me_const $= 9.10938356 \times 10^{-31}$	-1
Proton	mp_const $= 1.672621898 \times 10^{-27}$	+1
Neutron	mn_const $= 1.674927471 \times 10^{-27}$	0
Alpha particle	$6.6446573357 \times 10^{-27}$	+2
Positron	me_const $= 9.10938356 \times 10^{-31}$	+1

PARTICLE MASS

This section is shown when **Newtonian** or **Newtonian, first order** is selected as the **Formulation** and **User defined** is selected from the **Particle species** list.

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, $9.10938356 \times 10^{-31}$ 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** and **User defined** is selected from the **Particle species** list.

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 an option from the **Time dependence of field** list: **Stationary or time dependent** (the default), **Time harmonic**, or **Periodic**.

The default, **Stationary or time dependent**, does not modify the electric field when computing the force on the particles. This is appropriate for any of the following cases:

- An analytic expression for the electric field or potential has been specified,

- The field was computed using a Stationary study and should be treated as constant over time, or

- The field was computed using a Time Dependent study step, and there is a one-to-one correspondence between solution times for the field and for the particle trajectories.

Select **Time harmonic** when the field was computed using a Frequency Domain or Eigenfrequency study. The field is multiplied by a sinusoidal phase factor,

$$\mathbf{E} = \text{real}(\tilde{\mathbf{E}}\exp(j(\omega t + \phi_0)))$$

where $\mathbf{E}$ (SI unit: V/m) is the electric field value used to compute the force, $\tilde{\mathbf{E}}$ (SI unit: V/m) is the complex-valued electric field computed in the previous study, and ω (SI unit: rad/s) is the angular frequency. The angular frequency can be taken directly from the previous Frequency Domain or Eigenfrequency study, or it can be specified directly or in terms of the period.

Select **Periodic** when the field was computed using a previous Time Dependent study but is assumed to repeat over subsequent time intervals. This allows the trajectories of charged particles to be computed over many periods, while only having to solve for the electric field over a single period. The field is assumed to be periodic but is not required to oscillate sinusoidally over time; in this sense, the **Periodic** option is a generalization of the **Time harmonic** option.

The electric force is computed using the periodic electric field $\mathbf{E}_p$ (SI unit: V/m), which is related to the previously computed field by

$$\mathbf{E}_p(t) = \mathbf{E}(\text{mod}(t + \Delta t, T))$$

where t (SI unit: s) is the time, Δt (SI unit: s) is a user-defined time delay, and T (SI unit: s) is the period. Here mod is the modulo operator, which adds or subtracts a multiple of T from the first argument such that its value is between 0 and T.

For **Time harmonic** or **Periodic**, select an option from the **Frequency specification** list: **From solution** (the default), **Specify frequency**, or **Specify period**. For **Specify frequency**, enter the **Angular frequency** ω (SI unit: rad/s). The default is 1 MHz. For **Specify period**, enter the **Period** T (SI unit: s). The default is 1 μs.

For **Time harmonic**, also specify the **Initial phase angle** ϕ_0 (SI unit: rad). The default is 0.

For **Periodic**, also enter the **Time shift** Δt (SI unit: s). The default is 0.

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.

<table>
<tr><td>🔍</td><td>Electric Force Theory</td></tr>
</table>

Magnetic Force

<table>
<tr><td>📑</td><td>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</td></tr>
</table>

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 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.

<table>
<tr><td>▥</td><td>With the AC/DC Module see *Magnetic Lens*: Application Library path **ACDC_Module/Particle_Tracing/magnetic_lens**.</td></tr>
</table>

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** Φ and **Longitude** Θ. 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 an option from the **Time dependence of field** list: **Stationary or time dependent** (the default), **Time harmonic**, or **Periodic**.

The default, **Stationary or time dependent**, does not modify the magnetic flux density when computing the force on the particles. This is appropriate for any of the following cases:

- An analytic expression for the magnetic flux density has been specified,

- The field was computed using a Stationary study and should be treated as constant over time, or

- The field was computed using a Time Dependent study step, and there is a one-to-one correspondence between solution times for the field and for the particle trajectories.

Select **Time harmonic** when the field was computed using a Frequency Domain or Eigenfrequency study. The field is multiplied by a sinusoidal phase factor,

$$\mathbf{B} = \text{real}(\tilde{\mathbf{B}}\exp(j(\omega t + \phi_0)))$$

where $\mathbf{B}$ (SI unit: T) is the magnetic flux density used to compute the force, $\tilde{\mathbf{B}}$ (SI unit: T) is the complex-valued magnetic flux density computed in the previous study, and ω (SI unit: rad/s) is the angular frequency. The angular frequency can be taken directly from the previous Frequency Domain or Eigenfrequency study, or it can be specified directly or in terms of the period.

Select **Periodic** when the field was computed using a previous Time Dependent study but is assumed to repeat over subsequent time intervals. This allows the trajectories of charged particles to be computed over many periods, while only having to solve for the magnetic flux density over a single period. The field is assumed to be periodic but is not required to oscillate sinusoidally over time; in this sense, the **Periodic** option is a generalization of the **Time harmonic** option.

The electric force is computed using the periodic magnetic flux density $\mathbf{B}_p$ (SI unit: T), which is related to the previously computed field by

$$\mathbf{B}_p(t) = \mathbf{B}(\text{mod}(t + \Delta t, T))$$

where t (SI unit: s) is the time, Δt (SI unit: s) is a user-defined time delay, and T (SI unit: s) is the period. Here mod is the modulo operator, which adds or subtracts a multiple of T from the first argument such that its value is between 0 and T.

For **Time harmonic** or **Periodic**, select an option from the **Frequency specification** list: **From solution** (the default), **Specify frequency**, or **Specify period**. For **Specify frequency**,

enter the **Angular frequency** ω (SI unit: rad/s). The default is 1 MHz. For **Specify period**, enter the **Period** T (SI unit: s). The default is 1 μs.

For **Time harmonic**, also specify the **Initial phase angle** ϕ_0 (SI unit: rad). The default is 0.

For **Periodic**, also enter the **Time shift** Δt (SI unit: s). The default is 0.

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.

Q | 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. It defines an accumulated variable for a boundary heat source using the kinetic energy of the incident particles.

Q   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^2). 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**.

- **Molecular weight of surface material** M_{s} (SI unit: kg/mol), default 0.028 kg/mol.
- **Mass density of surface** ρ (SI unit: kg/m^3), default 2320 kg/m^3.

Q   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
- Particle-Matter Interaction Theory
- Particle Beam Theory
- Thermionic Emission Theory
- Specialized Boundary Accumulators
- References for the Charged Particle Tracing Interface

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,
- $e = 1.602176634 \times 10^{-19}$ 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:

$$\frac{d}{dt}(m_p\mathbf{v} + Ze\mathbf{A}) = Ze(\mathbf{v} \cdot \nabla)\mathbf{A} + Ze(\mathbf{v} \times \nabla \times \mathbf{A}) - Ze\nabla V \qquad (4\text{-}1)$$

The second term on the left-hand side of Equation 4-1 represents the total differential which can be expressed as:

$$\frac{d}{dt}(Ze\mathbf{A}) = Ze\left(\frac{\partial \mathbf{A}}{\partial t} + (\mathbf{v} \cdot \nabla)\mathbf{A}\right) \qquad (4\text{-}2)$$

Inserting Equation 4-2 into Equation 4-1 results in:

$$\frac{d}{dt}(m_p\mathbf{v}) = -Ze\frac{\partial \mathbf{A}}{\partial t} - Ze\nabla V + Ze(\mathbf{v} \times \nabla \times \mathbf{A}) \qquad (4\text{-}3)$$

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 becomes

$$\frac{d}{dt}(m_p\mathbf{v}) = -Ze\mathbf{E} + Ze(\mathbf{v}\times\mathbf{B}) \qquad (4-4)$$

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 an electric force $\mathbf{F}_e$ to the total force on the particles,

$$\mathbf{F}_e = Ze\mathbf{E}$$

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:

$$\mathbf{E} = -\nabla V$$

In the frequency domain, the electric field 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, ω is the angular frequency, ϕ_0 is the initial phase angle, and t is time.

Magnetic Force Theory

The Magnetic Force feature adds a magnetic force $\mathbf{F}_m$ to the total force on the particles,

$$\mathbf{F}_m = Ze(\mathbf{v}\times\mathbf{B})$$

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, ω is the angular frequency, ϕ_0 is the initial phase angle, and t is time.

Computing Earth's Magnetic Field

The **Magnetic Force** includes a model of Earth's magnetic field. The data used 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:

$$V(r, \Phi, \Theta) = R_e \sum_{l=1}^{L} \sum_{m=-l}^{l} \left(\frac{R_e}{r}\right)^{l-1} (g_l^m \cos(m\Phi) + h_l^m \sin(m\Phi)) P_l^m \cos(\Theta)$$

- r (SI unit: m) is the radial distance from Earth's center,
- L (dimensionless) is the maximum degree of the expansion,
- Φ (SI unit: rad) is the East longitude,
- Θ (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 node. Random collisional forces can be applied using the Collisions node.

FRICTION FORCE

The Friction Force feature adds the following contribution to $\mathbf{F}_t$:

$$\mathbf{F}_t = -m_p \nu(\mathbf{v} - \mathbf{u})$$

where $\mathbf{v}$ and $\mathbf{u}$ (SI unit: m/s) are the particle velocity and background fluid velocity. The collision frequency ν (SI unit: 1/s) is either specified directly, or via a collision cross section σ (SI unit: m^2) and background number density N_d (SI unit: $1/m^3$):

$$\nu = N_d \sigma |\mathbf{v} - \mathbf{u}|$$

> The limitations of the friction model are described in the Introduction to the Charged Particle Tracing Interface Theory section.

RANDOM COLLISIONAL FORCES

The Collisions node uses a stochastic approach to modeling particle collisions with the atoms or molecules in a background gas. Individual collisions between the model particles and the atoms or molecules in the background gas are detected by sampling random numbers from a distribution and comparing them to the collision probability, which usually varies over time. During a collision, the velocity of the model particle changes discontinuously. Depending on the type of reaction, secondary particles may be emitted as a result of the collision.

COLLISION DETECTION

A single **Collisions** node usually corresponds to a single species in the background gas. Add subnodes to the **Collisions** node to specify the different reaction types that can occur between the model particles and the background gas, such as elastic collisions and charge exchange reactions.

Derivation of Collision Probability

Consider a model particle interacting with a background gas with an average collision frequency $\nu(t)$ (SI unit: $1/s$) given by the expression

$$\nu(t) = \sum_i \nu_i(t)$$

where the $\nu_i(t)$ are the frequencies associated with individual reaction types between the model particle and each species in the background gas. For a time interval Δt (SI unit: s) much smaller than the average time between collisions, the probability $P_T(t)$ (dimensionless) that a collision takes place within a time interval of width Δt centered at a time t is

$$P_T(t) = \nu(t)\Delta t$$

Let $Q(t)$ (dimensionless) be the probability that a collision does not occur during the time interval $(0, t)$. Similarly, $Q(t + \Delta t)$ is the probability that a collision does not occur during the interval $(0, t + \Delta t)$. $Q(t + \Delta t)$ can be expressed the product of the probability that no collision occurs in $(0, t)$ (given by $Q(t)$), and furthermore that no collision occurs in the interval $(t, t + \Delta t)$. The latter can be expressed as $1 - P_T(t)$ or $1 - \nu(t)\Delta t$. Thus the relationship between $Q(t)$ and $Q(t + \Delta t)$ can be expressed as

$$Q(t + \Delta t) = Q(t)[1 - \nu(t)\Delta t]$$

Rearranging this equation yields

$$\frac{Q(t + \Delta t) - Q(t)}{\Delta t} = -\nu(t)Q(t)$$

Taking the limit as Δt approaches zero, the left-hand side becomes a time derivative,

$$\frac{dQ}{dt} = -\nu(t)Q(t)$$

Integrating and noting that $Q(0) = 1$ (that is, given no time, there is no possibility of a collision) yields

$$Q(t) = \exp\left(-\int_0^t \nu(t')dt'\right)$$

The probability that the next collision does occur in the interval $(0, t)$ is thus

$$P(t) = 1 - \exp\left(-\int_0^t \nu(t')dt'\right) \tag{4-5}$$

Predicting Whether and When a Collision Occurs

The **Collisions** node supports two algorithms for predicting the collision times. The algorithm is controlled by the **Collision detection** list in the settings window for the **Collisions** node. The two options are **At time steps taken by solver** and **Null collision method, cold gas approximation**.

The default option, **At time steps taken by solver**, can only detect and apply collisions at the discrete times used by the time dependent solver. At each one of these solution times, an uncorrelated, uniformly distributed random number U (dimensionless) is sampled from the interval $(0, 1)$ for each particle. The time step is assumed to be significantly smaller than the free time between collisions,

$$t \ll \frac{1}{\nu}$$

This assumption is important because the collision frequency is assumed constant over each time step taken by the solver. For constant collision frequency, Equation 4-5 reduces to

$$P(t) = 1 - \exp(-\nu t) \tag{4-6}$$

Since the collision frequency and time are always positive, it is clear that the collision probability tends toward zero as the time step becomes very small.

To detect whether a collision occurs, the collision probability for each particle is compared to the random number U,

$$U < 1 - \exp(-\nu t)$$

Because U is not used for any other purpose, and its actual value within the distribution has not yet been specified, we can take advantage of the equivalent probability distributions of U and $1 - U$ to simplify this expression slightly:

$$U > \exp(-\nu t)$$

When this option **At time steps taken by solver** is used, each model particle is only allowed to undergo at most one collision per time step taken by the solver. If the collision frequency is large enough so that particles are likely to undergo two or more collisions in a single time step taken by the solver, then the second and later collisions within the step are disregarded, potentially skewing the particle statistics.

Expressions for the Collision Frequency
In practice, the collision frequencies are usually not constant over time. In its most general form (see for example Ref. 3), the collision frequency for the jth reaction can be expressed in terms of the corresponding collision cross section σ_j (SI unit: m^2),

$$\nu_j(t) = n \int \sigma_j(|\mathbf{v} - \mathbf{v}_g|) |\mathbf{v} - \mathbf{v}_g| f(\mathbf{v}_g) d\mathbf{v}_g \tag{4-7}$$

where

- $\mathbf{v}_g$ (SI unit: m/s) is the velocity of an atom or molecule in the background gas,
- $\mathbf{v}$ (SI unit: m/s) is the particle velocity, and
- n (SI unit: $1/m^3$) is the number density of the gas.

The integral is taken over all velocity space. The probability distribution function $f(\mathbf{v}_g)$ (SI unit: s^3/m^3) is usually assumed to be a drifting Maxwell distribution,

$$f(\mathbf{v}_g) = \frac{1}{(2\pi k_B T/m_g)^{3/2}} \exp\left[-\frac{m_g|\mathbf{v}_g - \mathbf{u}|^2}{2k_B T}\right] \tag{4-8}$$

where

- m_g (SI unit: kg) is the mass of a gas molecule,
- $\mathbf{u}$ is the mean or drift velocity of the background gas,

- T (SI unit: K) is the gas temperature, and
- $k_B = 1.380649 \times 10^{-23}$ J/K is the Boltzmann constant.

Alternatively, the mass m_g can be expressed in terms of the molar mass M_g of the background species (SI unit: kg/mol),

$$m_g = \frac{M_g}{N_A}$$

where $N_A = 6.02214076 \times 10^{23}$ 1/mol is the Avogadro constant. Optionally, the molar mass can be assigned unique values or expressions for each reaction type.

For the option **At time steps taken by solver**, the following simplification is made to Equation 4-7 to remove the integration over velocity space. First, a gas molecule velocity is sampled at random from Equation 4-8, using a built-in `randomnormal` function that creates uncorrelated Gaussian pseudorandom numbers for each velocity component. Then the velocity distribution function in Equation 4-7 is replaced with a Dirac delta function at the sampled velocity components, yielding the simplified expression

$$\nu_j = n\sigma_j(\mathbf{v} - \mathbf{v}_g)|\mathbf{v} - \mathbf{v}_g| \tag{4-9}$$

where $\mathbf{v}_g$ now denotes the specific value of the background gas velocity sampled from the distribution.

The limiting time step size for the option **At time steps taken by solver** is determined from the following criteria:

1 The step size must be small enough so that the probability of a single model particle colliding with two or more background gas molecules in a single time step is negligibly small.

2 The collision frequency is dependent on particle velocity, both directly and due to the energy or velocity dependence of typical collision cross section data. The time step must be small enough so that the particle velocity can be treated as a constant value over the time step. This criterion is significant when the model particles are ions or electrons that can accelerate quickly in external electric fields.

In some cases, it is the number of collisions is so large, or the particles accelerate so rapidly, that it is impractical to specify a time step size small enough to satisfy both of these conditions. In such cases, the performance of the model may be significantly improved by using the null collision method described in the following section.

THE NULL COLLISION METHOD

The null collision method (Ref. 4) is one of a class of computational methods known as rejection methods. This approach is used by selecting **Null collision method, cold gas approximation** from the **Collision detection** list. The basic premise of the null collision method is to avoid inverting Equation 4-5 for an arbitrary functional form of $\nu(t)$ by first assigning a large, constant value ν_m to the collision frequency. A number of collision times are then sampled from a distribution based on this artificially large, constant collision frequency. The sampled collision times are then either used or rejected based on an integral of the real collision frequency $\nu(t)$ up to each trial time.

Cold Gas Approximation

The implementation of the null collision method in the **Collisions** node uses a simplifying assumption known as the cold gas approximation. In applications with extremely high particle speeds, such as ions or electrons accelerating in strong electric fields, the cold gas approximation can be used to simplify the expression for collision frequency. In the cold gas approximation, the background gas velocity is set to zero, so Equation 4-9 simplifies to

$$\nu_j = n\sigma_j(\mathbf{v})|\mathbf{v}|$$

The assumption is that the particle velocity is so much larger than the thermal velocity of the background gas molecules that the thermal velocity can be ignored entirely. Because the velocity distribution function of the background gas is no longer considered, the model particle is equally likely to collide with a molecule from any part of the distribution, an assumption that clearly breaks down then the particle velocity is comparable to the thermal velocity of the gas.

Trial Collision Frequency and Trial Times

As is the case with the option **At time steps taken by solver**, the first step to predicting the collision times is to generate a uniformly distributed random number U_1 for each particle and compare it to the collision probability over the time interval; we use the subscript 1 here because the null collision method requires multiple uncorrelated random numbers for each model particle. Instead of the simplification made in Equation 4-6, however, the random number is substituted directly into Equation 4-5,

$$U_1 = 1 - \exp\left(-\int_0^t \nu(t')dt'\right)$$

or equivalently,

$$U_1 = \exp\left(-\int_0^t v(t')dt'\right) \qquad (4\text{-}10)$$

The objective is to solve for t.

The next step is to assign a constant trial frequency v_m, such that $v_m > v(t)$ over the interval $(0, t)$. Substituting the corresponding trial time t_m for t, and v_m for $v(t)$, into Equation 4-10 and solving for t_m then yields

$$t_m = -\frac{1}{v_m}\log U_1$$

Rejecting Trial Collisions

After a trial time for the particle has been obtained, the next step is to check whether the collision detected at the trial time actually occurs. A second uniformly distributed random number U_2, not correlated with U_1, is also randomly sampled from the interval $(0, 1)$. If the inequality

$$U_2\int_0^{t_m} v_m dt > \int_0^{t_m} v(t)dt$$

holds true, then the collision is actually a null collision, and must be discarded. Because the trial frequency is constant, this inequality can be simplified as

$$U_2 > \frac{1}{v_m t_m}\int_0^{t_m} v(t')dt' \qquad (4\text{-}11)$$

Otherwise, the collision actually occurs.

In the event that a null collision occurs, the particle moves along its trajectory unaffected by the background gas until time t_m, when a new trial frequency and trial time are computed, and the process repeats itself.

There is often no analytic expression for the integral in Equation 4-11, since the functional form that relates the collision cross section and collision frequency is not known *a priori*. However, the integral can be approximated as

$$\int_0^{t_m} v(t')dt' \approx \frac{t_m}{2}[v(0) + 4v(t_m/2) + v(t_m)]$$

If the integral in Equation 4-11 were ever greater than unity, that would imply that the selected value of the trial frequency was too low and the process must be repeated, without advancing the particle in time to t_m.

After a real collision is detected, meaning that the trial frequency was sufficiently high but a null collision wasn't detected, the final step is to determine the type of collision to apply. If multiple types of collision are present, the probability p_j of a specific collision type occurring is

$$p_j = \frac{\nu_j}{\nu}$$

where the denominator is the total collision frequency over all collision types.

Maximum Number of Collisions
In principle, each model particle can undergo an arbitrarily large number of collisions in a single time step taken by the solver. In practice, however, this number is capped to ensure that the solver does not spend an inordinate amount of time in a single time step. This might happen, for example, if the time step is extremely large relative to the free time and the trial frequency is many orders of magnitude larger than the actual frequency at the beginning of the time step, since some particles might experience a large number of null collisions before any real collision.

The **Maximum number of consecutive null collisions** is available in the physics interface **Advanced Settings** section. The default is 100. If any particle undergoes more null collisions than the specified value, without undergoing any real collisions, then the particle cannot collide anymore until the next time step. When this happens, a Warning is produced in the solver sequence. Consider a modest reduction in the maximum time step size if this warning appears.

COLLISION TYPES

A single **Collisions** node corresponds to a single species in the background gas. Add subnodes to the **Collisions** node to specify the types of reaction that the model particles undergo with the background gas, such as elastic collisions and charge exchange reactions.

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:

$$\mathbf{v'} = \mathbf{v} - \frac{m_g}{m_p + m_g}(\mathbf{g} - \mathbf{g'})$$

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:

$$\mathbf{v'} = \mathbf{v} - \frac{m_g}{m_p + m_g}(\mathbf{g} - \mathbf{g'})$$

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 ΔE (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:

$$\mathbf{v'} = \mathbf{v} - \frac{m_g}{m_p + m_g}(\mathbf{g} - \mathbf{g'})$$

where $\mathbf{g'} = |\mathbf{g'}|\mathbf{R}$ and where

$$|\mathbf{g}'| = \sqrt{\mathbf{g}\cdot\mathbf{g} - \frac{2\Delta E(m_{\mathrm{p}} + m_{\mathrm{g}})}{m_{\mathrm{p}}m_{\mathrm{g}}}}$$

where $\mathbf{R}$ is a uniformly distributed random unit vector, and ΔE (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 node that is appropriate for electrons. This **Particle Properties** node should therefore have a mass (or rest mass) equal to me_const and a charge number of −1.

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 $\mathbf{v}_{\mathrm{n}}'$ are defined by the expressions:

$$\mathbf{v}' = \mathbf{v}_{\mathrm{g}} + \frac{m_{\mathrm{g}}}{m_{\mathrm{p}} + m_{\mathrm{g}}}(\mathbf{g} - \mathbf{g}')$$

$$\mathbf{v}_{\mathrm{n}}' = \mathbf{v} - \frac{m_{\mathrm{g}}}{m_{\mathrm{p}} + m_{\mathrm{g}}}(\mathbf{g} - \mathbf{g}')$$

where

$$\mathbf{g}' = \mathbf{g}_{\perp}\cos\chi + \mathbf{g}_{\|}\sin\chi$$

and χ (SI unit: rad) is the scattering angle in the center of mass coordinate system. The vectors $\mathbf{g}_{\|}$ and $\mathbf{g}_{\perp}$ are defined by the expressions:

$$\mathbf{g}_{\|} = \frac{|\mathbf{g}'|}{|\mathbf{g}|}\mathbf{g}$$

$$\mathbf{g}_{\perp} = |\mathbf{g}'|(\mathbf{e}_1\cos\phi + \mathbf{e}_2\sin\phi)$$

where ϕ (SI unit: rad) is a uniformly distributed angle between 0 and 2π. 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:

$$|\mathbf{g}'| = \sqrt{\mathbf{g} \cdot \mathbf{g} - \frac{2\Delta E(m_\mathrm{p} + m_\mathrm{g})}{m_\mathrm{p} m_\mathrm{g}}}$$

where ΔE (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:

$$\mathbf{F} = -S_\mathrm{e} \rho \frac{\mathbf{v}}{|\mathbf{v}|}$$

where

- $\mathbf{F}$ (SI unit: N) is the force on the ion,
- S_e (SI unit: $\mathrm{m}^4/\mathrm{s}^2$) is the electronic stopping power,
- ρ (SI unit: kg/m^3) is the mass density of the target material, and
- $\mathbf{v}$ (SI unit: m/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 χ (SI unit: rad) is computed for each particle using the expression

$$\chi = \pi - 2 \int_{\xi_{min}}^{\infty} \frac{b}{\xi^2 \sqrt{1 - \frac{\Phi(\xi)}{\xi\varepsilon} - \frac{b^2}{\xi^2}}} d\xi \qquad (4\text{-}12)$$

which is a dimensionless version of the expression for the scattering angle as given in Ref. 6, in which b (dimensionless) is the reduced impact parameter and ξ 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 4-3 below.

The reduced energy ε (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,
- $\varepsilon_0 = 8.854187817 \times 10^{-12}$ F/m is the permittivity of vacuum,
- $e = 1.602176634 \times 10^{-19}$ 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 ξ_{min} (dimensionless) is the largest positive root of the equation

$$-\xi_{min}^2 + \frac{\Phi(\xi_{min})}{\varepsilon}\xi_{min} + b^2 = 0$$

The screening function $\Phi(\chi)$ (dimensionless) changes depending on the option selected from the **Screening function** list. The available options are tabulated below.

TABLE 4-3: SCREENING LENGTH AND SCREENING FUNCTION DEFINITIONS

SCREENING FUNCTION	SCREENING LENGTH EXPRESSION a_I	SCREENING FUNCTION EXPRESSION $\Phi(\xi)$
None	$\dfrac{0.8853a_0}{Z_p^{0.23} + Z_m^{0.23}}$	1
Bohr	$\dfrac{a_0}{Z_p^{0.23} + Z_m^{0.23}}$	$\exp(-\xi)$
Moliere	$\dfrac{0.8853a_0}{Z_p^{2/3} + Z_m^{2/3}}$	$0.35\exp(-0.3\xi) + 5.5\exp(-1.2\xi) + 0.1\exp(-6\xi)$
Lenz-Jensen	$\dfrac{0.8853a_0}{Z_p^{2/3} + Z_m^{2/3}}$	$0.7466\exp(-1.038\xi) + 0.2433\exp(-0.3876\xi)$ $+ 0.01818\exp(-0.206\xi)$
Universal	$\dfrac{0.8853a_0}{Z_p^{0.23} + Z_m^{0.23}}$	$0.1818\exp(-3.2\xi) + 0.5099\exp(-0.9423\xi)$ $+0.2802\exp(-0.4028\xi) + 0.2817\exp(-0.2016\xi)$

Information on the various screening functions and supplemental references can be found in Ref. 7.

The value of the reduced impact parameter b is sampled from a Rayleigh probability distribution using the expression

$$b = \frac{1}{a_I}\sqrt{\frac{-\log(U)}{\pi LN}}$$

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 χ is very close to 0, so most collisions tend not to have a large effect on the particle trajectory. If the computed value of χ is less than the specified **Cutoff scattering angle** χ_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 ϕ 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

$$T = \frac{4m_1 m_2}{(m_1 + m_2)^2} E \sin\left(\frac{\chi}{2}\right)$$

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 4-12 at every time step for every particle, the value of this integral is tabulated for a range of values of b and ε, 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

$$-5 \leq \log(\varepsilon) \leq 15$$

$$-25 \leq \log(b) \leq 10$$

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. 8.

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, $\mathbf{q}_{av}$. In 2D, the transverse direction is orthogonal to the average particle velocity $\mathbf{v}_{av}$; in 3D, the two transverse directions are orthogonal to $\mathbf{v}_{av}$ 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 $\varepsilon_{1,rms}$ (SI unit: m) is defined as:

$$\varepsilon_{1,\,rms} = \sqrt{\langle x^2 \rangle \langle (x')^2 \rangle - \langle xx' \rangle^2}$$

Where the brackets represent an arithmetic mean over all particles. In addition, the 4-rms emittance $\varepsilon_{4,rms}$ is frequently reported because it corresponds to the area of an ellipse if the distribution of particles in phase space is uniform:

$$\varepsilon_{4,\,rms} = 4\varepsilon_{1,\,rms}$$

In addition to the size of the phase space ellipse, several characteristics of its shape can be described using the Twiss parameters α, β, and γ, defined as:

$$\alpha = -\frac{\langle xx' \rangle}{\varepsilon_{1,\,rms}}$$

$$\beta = \frac{\langle x^2 \rangle}{\varepsilon_{1,\,rms}}$$

$$\gamma = \frac{\langle (x')^2 \rangle}{\varepsilon_{1,\,\text{rms}}}$$

In 3D, it is also possible to define the hyperemittance:

$$\varepsilon_h = \varepsilon_1 \varepsilon_2$$

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 α 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 β and α.

Given the initial value of the Twiss parameter β, the 1-rms beam emittance ε and the Twiss parameter α, 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. 9:

$$A^2 = \left(\frac{x}{a}\right)^2 + \left(\frac{ax' - a'x}{\varepsilon_x}\right)^2 + \left(\frac{y}{b}\right)^2 + \left(\frac{by' - b'y}{\varepsilon_y}\right)^2$$

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. 9, 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:

$$x' = A_x\left(a'\cos(2\pi\hat{\beta}_x) - \frac{4\varepsilon_x}{a}\sin(2\pi\hat{\beta}_x)\right)$$

$$y' = A_y\left(b'\cos(2\pi\hat{\beta}_y) - \frac{4\varepsilon_y}{b}\sin(2\pi\hat{\beta}_y)\right)$$

$$z' = \sqrt{1 - x'^2 - y'^2}.$$

The definition of A changes depending on the requested distribution:

TABLE 4-4: DEFINITION OF VARIABLE A FOR DIFFERENT DISTRIBUTION TYPES

KV	WATERBAG	PARABOLIC
$A = 1$	$A = \sqrt{\frac{3}{2}\sqrt{\hat{u}_A}}$	$A = \sqrt{1 - 2\cos\left(\frac{\eta - 2\pi}{3}\right)}$
		$\eta = \mathrm{acos}(1 - 2\hat{u}_A)$

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

$$x' = \frac{a'}{a}x + \frac{4\varepsilon_x}{2a}\hat{g}_{x'}, \quad y' = \frac{b'}{b}y + \frac{4\varepsilon_y}{2b}\hat{g}_{y'}, \quad z' = \sqrt{1 - x'^2 - y'^2} .$$

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 β, Twiss parameter α and beam emittance ε 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^2) and the **Release current magnitude** I (SI unit: A) using

$$\varepsilon_{rms} = \sqrt{\frac{I}{B}} .$$

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**, θ using the following

$$\beta = \frac{x_m}{x'_m}$$

and

$$\varepsilon_{rms} = \frac{x_m x'_m}{4} .$$

The Twiss parameter α 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

$$x_{m,1} = \frac{x_m}{\sqrt{x_m^2 + x'^2_m}} \sqrt{\varepsilon_{rms}}$$

$$x'_{m,1} = \frac{x'_m}{\sqrt{x_m^2 + x'^2_m}} \sqrt{\varepsilon_{rms}}$$

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_{th} (SI unit: A/m^2) is (Ref. 10)

$$J_{th} = A^* T^2 \exp\left(-\frac{e\Phi}{k_B T}\right)$$

where

- T (SI unit: K) is the temperature of the cathode,
- Φ (SI unit: V) is the work function of the metal,
- A^* (SI unit: A/(m^2·K^2)) is the effective Richardson constant,

- $k_B = 1.380649 \times 10^{-23}$ J/K is the Boltzmann constant, and
- $e = 1.602176634 \times 10^{-19}$ C is the elementary charge.

The total current I_{th} (SI unit: A) released from the surface is

$$I_{th} = \iint_\Omega J_{th} d\Omega$$

where the surface integral is taken over the selected boundaries.

INITIAL VELOCITY DIRECTIONS

The initial particle velocity components are

$$v_{t1} = V \sin\theta \cos\varphi$$
$$v_{t2} = V \sin\theta \sin\varphi$$
$$v_n = V \cos\theta$$

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 φ is uniformly distributed in the interval $[0, 2\pi]$. The polar angle θ is

$$\theta = \operatorname{asin}\sqrt{U_1}$$

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

$$W = \frac{m_e}{2k_B T}V^2 \tag{4-13}$$

where $m_e = 9.10938356 \times 10^{-31}$ kg is the electron mass. The thermal electrons are assumed to be nonrelativistic. The probability distribution function of the normalized kinetic energy is

$$f(W) = W \exp(-W) \tag{4-14}$$

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 4-14 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 4-14 is

$$c(W) = \int_0^W W' \exp(-W')dW' = 1 - (W+1)\exp(-W)$$

which is already normalized, that is,

$$\lim_{W \to \infty}(1 - (W+1)\exp(-W)) = 1$$

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
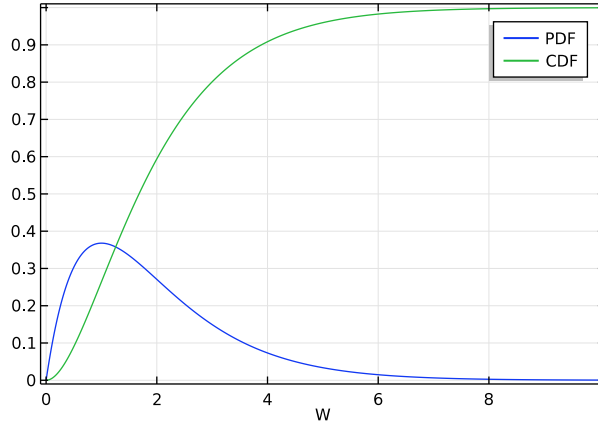
$$\text{LambertW}(-1, x)\exp(\text{LambertW}(-1, x)) = x$$

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 θ and azimuthal angle ϕ.

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 4-14, 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_{rel},

$$f_{rel} = \texttt{const}$$

To yield the specified total current I_{th}, the effective frequency of release of the ith model particle must be

$$f_{rel,\,i} = \frac{I_{th}}{eN}$$

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 4-14, the probability distribution must instead be incorporated into the effective frequency of release:

$$f_{rel} \propto W \exp(-W) \tag{4-15}$$

Thus the effective frequency of release of the ith particle is

$$f_{rel,i} = \frac{I_{th}}{e} \frac{W_i \exp(-W_i)}{\sum W_j \exp(-W_j)}$$

where the sum over j in the denominator is taken over all model particles released by the feature.

Equation 4-14 has a local maximum at $W = 1$, after which it gradually decreases. Some representative values of the CDF are given in Table 4-5.

TABLE 4-5: SELECTED VALUES OF THE CDF FOR NORMALIZED ELECTRON ENERGY

W	CDF
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 $[0, V_{max}]$, where

$$V_{max} = n \sqrt{\frac{2 k_B T}{m_e}}$$

where n (dimensionless) is a user-defined proportionality factor.

Differentiating both sides of Equation 4-13 and rearranging yields

$$dV = \sqrt{\frac{k_B T}{2 m_e W}} dW$$

Substituting into Equation 4-14 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 4-15 the proportionality factor is multiplied by an additional term proportional to $\sqrt{W}$:

$$f_{rel} \propto W^{3/2}\exp(-W) \tag{4-16}$$

Thus the frequency of release of the ith particle is

$$f_{rel,i} = \frac{I_{th}}{e}\frac{W_i^{3/2}\exp(-W_i)}{\sum W_j^{3/2}\exp(-W_j)}$$

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.

Specialized Boundary Accumulators

When particles interact with a surface, it is possible for them to contribute to one or more dependent variables defined on the surface. Such variables, called accumulated

variables, are described in detail in Accumulator Theory: Boundaries in Theory for the Mathematical Particle Tracing Interface.

The following specialized accumulators are available for the Charged Particle Tracing interface. In general, these accumulators are of the density type. When a particle hits a boundary mesh element, the value of the accumulated variable in that element rpb is incremented as follows:

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

where rpb_{new} is the post-collision value of the accumulated variable, rpb is the pre-collision value, R is a source term, and V is the mesh element length (in 2D) or surface area (in 3D). Each of the accumulated variables in this section is defined by the physical quantity that it represents and the expression for its source term.

SURFACE CHARGE DENSITY THEORY

The Surface Charge Density feature adds an accumulated variable for the charge density σ (SI unit: C/m^2) on a set of boundaries that the particles hit. The source term for this accumulator type is

$$R = eZ$$

where $e = 1.602176634 \times 10^{-19}$ C is the elementary charge and Z (dimensionless) is the charge number of the incident particle.

CURRENT DENSITY THEORY

The Current Density feature adds an accumulated variable for the current density of the incident particles on a set of boundaries.

If the **Type** is set to **Normal current density**, the accumulated variable is the normal current density J_n (SI unit: A/m^2), and the source term is

$$R = f_{\text{rel}} e Z \cos\phi$$

where

- $e = 1.602176634 \times 10^{-19}$ C is the elementary charge,
- Z (dimensionless) is the charge number of the incident particle,

- ϕ (SI unit: rad) is the acute angle of incidence measured from the surface normal, and
- f_{rel} (SI unit: 1/s) is the effective frequency of release; that is, the number of real charged particles per unit time that are represented by each model particle.

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,

$$(f_{rel})_i = \frac{I}{N_{tf}eZ}$$

If the **Type** is set to **Current density**, one accumulated variable is defined for each component of the current density vector $\mathbf{J}_s$. The source term for the ith component is

$$R_i = f_{rel}eZ\frac{v_i}{|\mathbf{v}|}$$

where $\mathbf{v}$ (SI unit: m/s) is the incident particle velocity.

HEAT SOURCE THEORY

The Heat Source feature adds an accumulated variable for a heat source Q (SI unit: W/m^2) equal to the total kinetic energy of the incident particles. The source term is

$$R = f_{rel}E$$

where f_{rel} (SI unit: 1/s) is the effective frequency of release of the incident particle and E (SI unit: J) is the kinetic energy of the incident particle.

ETCH THEORY

The Etch feature uses accumulated variables to calculate the etch rate due to physical sputtering as energetic ions hit 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. 11) and by Guo and Sawin (Ref. 12).

If the **Plasma type** is **Collisionless**, then the number of real ions represented by each model particle can be deduced directly from the particle release feature via the effective frequency of release. A single accumulated variable E_E is defined for the etch rate on the surface (SI unit: m/s). The corresponding source term is

$$E_{E,i} = \frac{M_s}{\rho N_A} f_{rel} Y(\phi) \left(\sqrt{\frac{E}{Y_s}} - \sqrt{\frac{E_{th}}{Y_s}} \right)$$

where

- M_s (SI unit: kg/mol) is the molar mass of the surface material,
- ρ (SI unit: kg/m^3) is the density of the surface material,
- $N_A = 6.02214076 \times 10^{23}$ 1/mol is Avogadro's constant,
- f_{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 ϕ,
- E (SI unit: J) is the particle kinetic energy,
- E_{th} (SI unit: J) is the threshold energy, and
- Y_s (SI unit: J) is the slope of the etch yield curve.

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 Γ_i (SI unit: 1/(m^2·s)), defined as

$$\Gamma_i = \frac{\mathbf{n} \cdot \mathbf{J}_i}{\sum eZ \sin\phi}$$

where the numerator $\mathbf{n} \cdot \mathbf{J}_i$ (SI unit: A/m^2) is the normal current of charged particles at the surface. Usually the incident current from a collisional plasma is computed using the Plasma Module. The sum in the denominator is taken over all particles that hit the ith boundary mesh element. The etch rate in this element 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. 12) 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 ϕ 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/IAGA/vmod/igrf.html.

3. Z. Ristivojevic and Z.L. Petrovic. "A Monte Carlo simulation of ion transport at finite temperatures." *Plasma Sources Science and Technology*, vol. 21, no. 3, 035001, 12 pp, 2012.

4. H.R. Skullerud, "The stochastic computer simulation of ion motion in a gas subjected to a constant electric field." *Journal of Physics D: Applied Physics*, vol. 1, no. 11, pp. 1567-1568, 1968.

5. K. Nanbu, "Probability theory of electron-molecule, ion-molecule, molecule-molecule, and Coulomb collisions for particle modeling of materials processing plasmas and cases." *IEEE Transactions on plasma science*, vol. 28, no. 3, pp. 971-990, 2000.

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

7. J.F. Ziegler, J.P. Biersack, and M. D. Ziegler, *The Stopping and Range of Ions and Matter*, SRIM, 2015.

8. S. Humphries, *Charged Particle Beams*, Dover, 2013.

9. 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.

10. P.T. Kirstein, G. S. Kino, and W. E. Waters, *Space charge flow*, McGraw-Hill, 1967.

11. 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.

12. 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.

5

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 (fpt)** interface (⚛), found under the **Fluid Flow>Particle Tracing** branch (⚛) when adding a physics interface, is used to simulate the motion of particles in a background fluid. Particle motion can be driven by a combination of forces including drag, gravity, electric, magnetic, acoustophoretic, and user-defined forces. It is also possible to specify particle size or mass distributions, solve for particle temperature, and model two-way coupled 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 `fpt`.

PARTICLE RELEASE AND PROPAGATION

The **Formulation**, **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. The **Relativistic Correction** check box is not 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.

Particle Size Distribution

Select an option from the **Particle size distribution** list: **Uniform size** (the default), **Specify particle mass**, or **Specify particle diameter**.

When **Uniform size** is selected, the size of each particle is controlled by the settings for the Particle Properties node. The size of each particle is assumed constant over time, unless its diameter or mass is defined as an explicit function of time. Note that you can still release particles of different sizes in one model by creating multiple instances of the **Particle Properties** node.

When **Specify particle mass** (or **diameter**) is selected, you can specify the initial value of the mass (or diameter) in the settings for particle release features such as Inlet and Release from Grid. Optionally, you may release particles with a nonuniform size distribution. Particles may also grow or shrink over time, after they have been released; in the settings for the **Particle Properties** node, enter an expression for the **Accretion rate** R (SI unit: kg/s), which is the time derivative of the particle mass. Because the particle mass (or diameter) is solved for, these settings each use one additional degree of freedom per particle, compared to the **Uniform size** option.

One of the options **Specify particle mass** or **Specify particle diameter** 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 or Diameter 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.

⚠️	Adjusting certain physics interface settings, such as **Include out-of-plane degrees of freedom**, **Particle size distribution**, 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.

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

Q
- 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 2-4 for links to common sections and Table 2-5 to common feature nodes. You can also search for information: press F1 to open the **Help** window or Ctrl+F1 to open the **Documentation** window.

	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 Radiation Force
- Boundary Load
- Brownian Force
- Charge Accumulation
- Convective Heat Losses
- Dielectrophoretic Force
- Drag Force
- Droplet Breakup
- Electric Force
- Erosion
- Gravity Force
- Heat Source
- Kelvin-Helmholtz Breakup Model
- Lift Force
- Magnetic Force

- Magnetophoretic Force
- Mass Deposition
- Mass Flux
- Nozzle
- Nozzle Domain
- Charge Accumulation
- Particle Properties
- Radiative Heat Losses
- Rayleigh-Taylor Breakup Model
- 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)
- Accumulator (for Velocity Reinitialization)
- Auxiliary Dependent Variable
- Axial Symmetry
- Force
- Inlet
- Nonlocal Accumulator
- Outlet
- 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 size, density, 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 **Uniform size** is selected from the **Particle size distribution** list in the 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** ρ_p (SI unit: kg/m^3). The default is 2,200 kg/m^3.
- **Particle diameter** d_p (SI unit: m). The default is 1 μm.
- **Particle mass** m_p (SI unit kg). The default is 1 mg.

If **Specify particle diameter** or **Specify particle mass** is selected from the **Particle size distribution** list, just enter a value or expression for the **Particle density**. Then the initial particle mass or diameter is controlled by the release features.

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** μ_p (SI unit: Pa·s) the default is 1 mPa·s.
- **Particle surface tension** σ_p (SI unit: N/m). The default is 0.0729 N/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. The particle charge is the product of the charge number with the elementary charge e.

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: m/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 is $2,200$ J/(kg·K). Then enter a value or expression for the **Particle latent heat** h_p (SI unit: J/kg). The default value is $42,000$ J/kg.

ACCRETION RATE

This section is shown when **Specify particle diameter** or **Specify particle mass** is selected from the **Particle size distribution** list 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. If this value is positive, particles will grow continuously over time after they are released.

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 exert a drag force on particles in a fluid.

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 nonspherical, 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 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** μ (SI unit: Pa·s) is taken **From material**. For **User defined** enter another value or expression.

- For all choices, the fluid **Density** ρ (SI unit: kg/m^3) 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** μ_p (SI unit: Pa·s). The default value is 0.018 mPa·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** λ'/λ (dimensionless). The default value is 1.

- If **Basset**, **Epstein**, or **Phillips** is selected from the **Rarefaction effects** list, enter an **Accommodation coefficient** σ_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. When entering a set of Cunningham coefficients, note that the COMSOL implementation of the Cunningham correction factor defines the relative Knudsen number using particle diameter, not radius.

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 in a Rarefied Flow in Theory for the Particle Tracing for Fluid Flow Interface.

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.

❗	In 3D models, selecting the check box **Include wall corrections** can cause a significant increase in computation time, due to the additional overhead of locating the nearest point on a wall for each particle at every time step. This is most noticeable if the boundary mesh is very fine.

🔍	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**, 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 term is held constant for a time interval equal to the interaction time of the particle with an eddy in the flow.

- 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** or **Continuous random walk** is selected, enter values or expressions for the following:

- The **Turbulent kinetic energy** k (SI unit: m^2/s^2) 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.

- The **Turbulent dissipation rate** ε (SI unit: m^2/s^3) is related to the lifetime of eddy currents in the flow. If a physics interface is present that computes the turbulent dissipation rate, it can be selected directly from the list.

- The **Lagrangian time scale coefficient** C_L (dimensionless) is used to compute the Lagrangian time scale of the particle-eddy interactions. The default value is 0.2.

The **Continuous random walk** model also supports corrections for anisotropic turbulence, which is prevalent in wall-bounded flows. 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.

| ! | In 3D models, selecting the check box **Include anisotropic turbulence in boundary layers** can cause a significant increase in computation time, due to the additional overhead of locating the nearest point on a wall for each particle at every time step. This is most noticeable if the boundary mesh is very fine. |

| 🔍 | Particle Motion in a Turbulent Flow in Theory for the Particle Tracing for Fluid Flow Interface. |

ADDITIONAL TERMS

The **Include virtual mass and pressure gradient forces** check box is cleared by default. Select this check box to exert additional forces on particles called the virtual mass (or added mass) force and the pressure gradient force. The virtual mass force is exerted on a particle moving through a fluid, as fluid must be displaced to fill the empty space the particle leaves behind.

The virtual mass and pressure gradient forces are most significant when the density of the surrounding fluid is of a similar order of magnitude to (or greater order of magnitude than) the particle density. Therefore they are often neglected when modeling solid particles in a gas, but can be very significant when modeling solid particles in a liquid. These forces also depend on the temporal and spatial derivatives of the fluid velocity components. Consider increasing the discretization order of the degrees of freedom for fluid velocity when modeling the fluid flow.

🔍	Virtual Mass Force and Pressure Gradient Force 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.

❗	In 3D models, using the **Wall induced** lift force can cause a significant increase in computation time, due to the additional overhead of locating the nearest point on a wall for each particle at every time step. This is most noticeable if the boundary mesh is very fine.

Enter coordinates for the **Velocity field 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** μ (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 force accounts for the random motion of molecules in the vicinity of the particle. The **Brownian Force** is most significant for sub-micron particles and can usually be ignored for larger particles.

> When the **Brownian Force** is included in a model, the **Update scaled absolute tolerance** check box is cleared in the **Time-Dependent** solver settings. This prevents the solver from taking extremely small time steps to try to compensate for the random nature of the force.

BROWNIAN FORCE

Enter a value or expression for the **Dynamic viscosity** μ (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 a **Parameters** (P_i) node. Or in the **Model Builder**, right-click **Global Definitions** (🌐) and add a **Parameters** node.

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 nonphysical behavior, consider selecting **1** from the **Wall accuracy order** list when modeling Brownian motion.

	• About Parameters, Variables, Variable Utilities, and Expressions and Parametric Sweep in the *COMSOL Multiphysics Reference Manual*. • Brownian Force 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 exert a gravitational force on particles. By specifying the density of the surrounding fluid, the effect of buoyancy on the particle motion can also be included. The gravity vector can point in any direction with any magnitude, although the default is for particles to move downward using the acceleration due to gravity at the Earth's surface. For submicron particles, the drag and other external forces can dominate, so gravity can often have little effect on the particle trajectories.

GRAVITY FORCE

Enter coordinates based on space dimension for the **Gravity vector g** (SI unit: m/s^2). The default magnitude for the gravity vector is g_const, which is a built-in physical constant equal to $g = 9.80665$ m/s^2 corresponding to the standard acceleration due

to gravity on Earth. The default direction is the negative *y*-direction in a 2D geometry or the negative *z*-direction for a 2D axisymmetric or 3D geometry.

The default **Density** ρ (SI unit: 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 Radiation Force

📋	The **Acoustophoretic Radiation 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 Radiation Force** node to exert forces on small particles due to acoustic radiation. To be subjected to this force, particles must be in a region where the acoustic pressure or acoustic velocity is not spatially uniform.

In addition, the **Acoustophoretic Radiation Force** is only applied if the acoustic pressure or acoustic velocity has already been computed in a **Frequency Domain** study type. This is because the magnitude of the acoustophoretic force depends on the angular frequency.

RADIATION FORCE MODEL

This section determines how the equation for the acoustophoretic radiation force is defined. It also determines what inputs will be shown in subsequent sections.

Select an option from the **Particle type** list: **Solid particle** (the default) or **Liquid droplet**.

Select an option from the **Thermodynamic loss model**: **Ideal** (the default), **Viscous**, or **Thermoviscous**. The **Thermoviscous** loss model is the most detailed of the three but requires the most in-depth knowledge of the particle and fluid material 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.

- **Acoustic velocity** $\mathbf{u}$ (SI unit: m/s). Similarly, this should be solved for in another physics interface such as the Pressure Acoustics, Frequency Domain interface.

PARTICLE MATERIAL PROPERTIES

Enter the material properties of the solid particles or liquid droplet in this section. Some additional particle properties, such as density, are instead specified in the settings for the Particle Properties node.

For all thermodynamic loss models, if the **Particle type** is **Solid particle**, enter values or expressions for the following:

- **Pressure-wave speed** $c_{p,p}$ (SI unit: m/s, default 2,400 m/s), and

- **Shear-wave speed** $c_{s,p}$ (SI unit: m/s, default 1150 m/s).

If instead the **Particle type** is **Liquid droplet**, just enter the **Adiabatic speed of sound** c_p (SI unit: m/s, default 1445 m/s).

If the **Thermodynamic loss model** is **Viscous** or **Thermoviscous**, and the **Particle type** is **Liquid droplet**, also enter the **Particle dynamic viscosity** μ_p (SI unit: Pa·s, default 4.2×10^{-3} Pa·s).

If the **Thermodynamic loss model** is **Thermoviscous** and the **Particle type** is **Liquid droplet**, also enter the **Bulk viscosity particle** $\mu_{B,p}$ (SI unit: Pa·s, default 0).

If the **Thermodynamic loss model** is **Thermoviscous**, for either **Particle Type** enter the:

- **Heat capacity at constant pressure** $C_{p,p}$ (SI unit: J/(kg·K), default 2.1×10^3 J/(kg·K)),

- **Isobaric coefficient of thermal expansion** $\alpha_{p,p}$ (SI unit: 1/K, default 7.05 1/K,)

- **Ratio of specific heats** γ_p (dimensionless, default 1.15), and

- **Thermal conductivity** k_p (SI unit: W/(m K), default 0.17 W/(m K)). The thermal conductivity is always assumed to be a scalar for the purpose of computing the acoustophoretic radiation force.

SURROUNDING FLUID PROPERTIES

All of the inputs in this section are domain material properties that can either be taken **From material** or given a **User defined** expression. The defaults given below are shown by first selecting **User defined**.

For any **Thermodynamic loss model**, enter the:

- **Density** ρ (SI unit: kg/m^3, default 995 kg/m^3), and
- **Speed of sound** c (SI unit: m/s, default 1500 m/s).

If the **Thermodynamic loss model** is **Viscous** or **Thermoviscous**, also enter the **Dynamic viscosity** μ (SI unit: Pa·s, default 8.4×10^{-4} Pa·s).

If the **Thermodynamic loss model** is **Thermoviscous**, also enter the:

- **Bulk viscosity** μ_B (SI unit: Pa·s, default 2.4×10^{-3} Pa·s),
- **Heat capacity at constant pressure** C_p (SI unit: J/(kg·K), default 4.18×10^3 J/(kg·K)),
- **Isobaric coefficient of thermal expansion** α_p (SI unit: 1/K, default 2.75×10^{-4} 1/K),
- **Ratio of specific heats** γ (dimensionless, default 1.01), and
- **Thermal conductivity** k (SI unit: W/(m·K), default 0.61 W/(m·K)). The thermal conductivity is always assumed to be a scalar for the purpose of computing the acoustophoretic radiation force.

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.

Note on Backward Compatibility

In previous versions, the **Acoustophoretic Radiation Force** was called the **Acoustophoretic Force**, and it was possible to specify the particle bulk modulus K_p (SI unit: Pa) or compressibility β_p (SI unit: 1/Pa) directly, instead of the speed of sound. However, now the particle speed of sound is always required, and these other inputs are not shown.

If a model built in a previous version is opened in the current version, then the speed of sound will be defined as follows to preserve backward compatibility, where `<name>` is the unique identifier for the physics interface (probably `fpt`):

- If the old expression for the particle bulk modulus was `<expr>`, the new expression for the compressional speed of sound is `sqrt(<expr>/<name>.rhop)`.

- If the old expression for the particle compressibility was `<expr>`, the new expression for the compressional speed of sound is `sqrt(1/(<expr>*<name>.rhop))`.

<table>
<tr><td>🔍</td><td>

- Studies and Solvers and Frequency Domain in the *COMSOL Multiphysics Reference Manual*
- Acoustophoretic Radiation Force in the theory section.

</td></tr>
</table>

Electric Force

<table>
<tr><td>📝</td><td>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.</td></tr>
</table>

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.

<table>
<tr><td>🔍</td><td>Electric and Magnetic Forces in the theory section.</td></tr>
</table>

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 no 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.

Electric and Magnetic 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. Even an uncharged particle is subjected to a force in a nonuniform electric field if the particle's permittivity is different than the permittivity of the surrounding fluid.

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 electric field norm is greater. When the particle relative permittivity is less than the relative permittivity of the fluid, the particles are attracted to regions where the electric field norm is smaller.

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 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** $\varepsilon_{r,p}$ (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** σ_p (SI unit: S/m). The default value is 0.

FLUID PROPERTIES

By default the **Relative permittivity** ε_r (dimensionless) is taken **From material**. Select **User defined** from the list to enter a value or expression.

By default the **Electrical conductivity** σ (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** ω (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*
- Dielectrophoretic Force 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. In other words, use the **Shell** subnode to model dielectrophoresis of spherical particles with thin outer membranes.

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 0.1 μm. Enter a value for the **Shell relative permittivity** $\varepsilon_{r,s}$ (dimensionless). The default is 1. Enter a **Shell electrical conductivity** σ_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 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 Magnetophoretic Force 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.

> **Q** Magnetophoretic Force 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 exert the thermophoretic force on particles. The phenomenon called thermophoresis causes particles to migrate in a fluid with a nonuniform temperature field.

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. This is important when the fluid is a rarefied gas or the particles are extremely small. 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** μ (SI unit: Pa·s). The default is 0.02 mPa·s.
- **Density** ρ (SI unit: kg/m^3). The default is 1 kg/m^3.
- **Thermal Conductivity** k (SI unit: W/(m·K)). For **User defined** select **Isotropic**, **Diagonal**, **Symmetric**, or **Full** and then enter values in the field or matrix.
- **Heat capacity at constant pressure** C_p (SI unit: J/(kg·K)). The default is 1 kJ/(kg·K).

If the **Waldmann**, **Talbot**, or **Linearized BGK** model is used, also 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** λ'/λ 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** α_E (dimensionless) and a **Momentum accommodation coefficient** α_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.

Ⓠ   Thermophoretic Force 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: kg/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: kg/m^2/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** Δm_i (SI unit: kg). The default is 10^{-15} kg.

- If **Specify release times** is selected from the **Particle release specification** list, enter the **Erosive wear of particle** $E_{M,i}$ (SI unit: kg/s). The default is 10^{-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 GPa.

- **Mass density of surface** ρ (SI unit: kg/m^3). The default is 7,500 kg/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: m^3/kg). The default is 65 mm^3/kg.

- **Reference diameter** d_{ref} (SI unit: m). The default is 326 μm.

- **Reference velocity** v_{ref} (SI unit: m/s). The default is 104 m/s.

- Each of the following dimensionless **Oka model coefficients**:

  - 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^2). The default is 2 GPa.

- **Mass density of surface** ρ (SI unit: kg/m^3). The default is 7,500 kg/m^3.

DNV

- **DNV model coefficient** K (dimensionless). The default is 2×10^{-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**

⊕	Erosion Theory in the theory section.

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** (the default) 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** (the default). 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 volume force 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. This is equivalent to the assumption that the particle Biot number is much smaller than unity, and allows each particle's temperature to be stored as a single number instead of a temperature distribution.

CONVECTIVE HEAT LOSSES

Enter a **Heat transfer coefficient** h (SI unit: W/(m^2 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.

> Q   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** ε_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.

> Q   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 user-defined source or sink 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.
- Either **Specify particle diameter** or **Specify particle mass** must be selected from the **Particle size distribution** list in the physics interface **Additional Variables** section.
- **Liquid droplets/bubbles** must be selected from the **Particle type** list in the settings window for at least one Particle Properties node.
- 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** ρ (SI unit: kg/m^3) is taken **From material**. For **User defined** enter another value or expression. The default is 1 kg/m^3.

Enter the components of the **Velocity field 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.

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.
- Either **Specify particle diameter** or **Specify particle mass** must be selected from the **Particle size distribution** list in the physics interface **Additional Variables** section. Furthermore, the **Enable macroparticles** check box must be selected.
- **Liquid droplets/bubbles** must be selected from the **Particle type** list in the settings window for at least one Particle Properties node.

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: **Released Particle Properties**, **Initial Value of Auxiliary Dependent Variables**, and **Advanced Settings**.

RELEASE TIMES
Enter values for the **Release times** (SI unit: s). By default the text field contains the times 0 and 1 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 g/s.

For **From change in pressure** enter a value or expression for the **Change in pressure** Δp (SI unit: N/m^2). The default value is 1 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: N/m^2). The default value is 1 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 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 (![range icon]) 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:

- $q_{x,0}$ = range(0,1,3)
- $q_{y,0}$ = 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 text 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 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** Θ (SI unit: rad). The default value is $10°$.

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** ρ (SI unit: 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: N/m^2). 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.

Charge Accumulation

Use the **Charge Accumulation** node to compute the charge accumulation on a particle immersed in a gas background with free ions as a function of time.

🔍	Charge Accumulation in the theory section.

Enter a value or expression for the **Temperature** T (SI unit: K). The default value is 293.15 K. If a physics interface that computes the gas temperature is present, then this can be selected from the list as a model input.

Enter a value or expression for the **Pressure** p (SI unit: Pa). The default value is 1 atm. If a physics interface that computes the gas pressure is present, then this can be selected from the list as a model input.

The gas temperature and pressure are used to compute the gas number density, which is used to compute the ion mobility from the given ion reduced mobility.

PARTICLE CHARGING MODEL

Select a **Particle charging Model**: **Lawless** (the default), **Classical diffusion**, **Classical field**, **Classical diffusion and Field**, or **White**.

INITIAL ACCUMULATED CHARGE NUMBER

Enter a value for the **Initial accumulated charge number** $Z_{a,j}$ (dimensionless). The default is 0.

PARTICLE PROPERTIES

This section is only available when one of the following options is selected from the **Particle charging model** list: **Lawless**, **Classical field** or **Classical diffusion and Field**.

Select as option from the **Particle electric properties** list: **Dielectric** (the default), or **Conductive**. If **Dielectric** is selected enter a value for the **Particle relative permittivity** $\varepsilon_{r,p}$ (dimensionless) the default value is 1.

ION PROPERTIES

Enter a value or expression for the **Space charge density** ρ_q (SI unit: C/m^3). The default value is 10^{-5} C/m^3. If a physics interface that computes the space charge density is present, then this can be selected from the list.

If the charging model **Lawless**, **Classical diffusion**, **Classical field**, or **Classical diffusion and Field** is selected from the **Particle charging model** list, enter a value or expression for the **Reduced ion mobility** $\mu_i N$ (SI unit: $1/(Vms)$). The default value is 3×10^{21} $1/(Vms)$.

Enter a value or expression for the **Ion Temperature** T_i (SI unit: K). The default value is 293.1 K.

If the charging model **White** is selected from the **Particle charging model** list enter a value or expression for the **Ion Mass** M_i (SI unit: kg/mol). The default is 100 g/mol.

ELECTRIC FIELD

This section is only available when one of the following options is selected from the **Particle charging model** list: **Lawless**, **Classical field** or **Classical diffusion and Field**.

Select an option from the **Specify force using** list: **Electric potential** (the default) or **Electric field**.

- For **Electric potential** enter a value or expression for the **Electric potential** V (SI unit: V). If the electric potential is computed by another physics interface then it can be selected from the list.

- 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.

The Droplet Sprays in Fluid Flow Interface

The **Droplet Sprays in Fluid Flow (fpt)** 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:

- **Specify particle mass** is selected from the **Particle size distribution** list 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

The motion of particles in a fluid follows Newton's second law, which states that the net force on an object is equal to the time derivative of its linear momentum in an inertial reference frame:

$$\frac{d}{dt}(m_p \mathbf{v}) = \mathbf{F}_D + \mathbf{F}_g + \mathbf{F}_{ext}$$

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

(5-1)

where

- m_p (SI unit: kg) is the particle mass,
- $\mathbf{v}$ (SI unit: kg) is the particle velocity,

- **q** (SI unit: m) is the particle position, and
- **F**$_D$, **F**$_g$, and **F**$_{ext}$ (SI unit: N) are the drag, gravity, and other forces, respectively.

In the COMSOL implementation, when the particle mass is solved for as an additional degree of freedom, such that accretion or evaporation can take place, the mass is moved outside the time derivative to prevent nonphysical acceleration of the particles:

$$m_p \frac{d\mathbf{v}}{dt} = \mathbf{F}_D + \mathbf{F}_g + \mathbf{F}_{ext}$$

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.

If the *virtual mass* (or *added mass*) term is applied in addition to the drag force, then the particle mass m_p on the left-hand side is replaced by the virtual particle mass m_v. For more information about the virtual mass term, see the Virtual Mass Force section.

THE STOKES DRAG LAW

In Equation 5-1, the Drag Force **F**$_D$ is defined as:

$$\mathbf{F}_D = \left(\frac{1}{\tau_p}\right) m_p (\mathbf{u} - \mathbf{v}) \qquad (5\text{-}2)$$

where

- τ_p is the particle velocity response time (SI unit: s)
- **v** is the velocity of the particle (SI unit: m/s), and
- **u** is the fluid velocity (SI unit: m/s) at the particle's position.

Strictly speaking, **u** is the value that the fluid velocity would have at the particle's position if no particle were there, unless fluid-particle interactions are considered.

Many expressions for τ_p are available. The validity of a given drag law depends on the relative Reynolds number Re_r (dimensionless) of particles in the flow,

$$Re_r = \frac{\rho |\mathbf{u} - \mathbf{v}| d_p}{\mu}$$

where

- d_p (SI unit: m) is the particle diameter,
- ρ (SI unit: kg/m^3) is the density of the fluid, and
- μ (SI unit: Pa·s) is the dynamic viscosity of the fluid.

Many expressions for the drag force, and the range of relative Reynolds numbers at which they are applicable, are given in Ref. 4.

In a creeping flow, where the relative Reynolds number is very low ($\text{Re}_r \ll 1$), the Stokes drag law is applicable. This is the default behavior when adding a **Drag Force** node to the Particle Tracing for Fluid Flow interface. In the Stokes drag law, the velocity response time is defined as

$$\tau_p = \frac{\rho_p d_p^2}{18\mu} \tag{5-3}$$

where

- μ is the fluid viscosity (SI unit: Pa·s),
- ρ_p is the particle density (SI unit: kg/m^3), and
- d_p is the particle diameter (SI unit: m).

By substituting Equation 5-3 into Equation 5-2, some other familiar expressions for the Stokes drag force can be obtained,

$$\mathbf{F}_D = 3\pi\mu d_p(\mathbf{u}-\mathbf{v}) = 6\pi\mu r_p(\mathbf{u}-\mathbf{v})$$

where r_p (SI unit: m) is the particle radius.

OTHER DRAG LAWS

The Stokes drag law is not applicable at larger relative Reynolds numbers. Therefore, other drag laws are needed when the particles are very large, when their velocity relative to the fluid is very fast, when the fluid viscosity is low, or some combination of these factors.

To simplify the expressions for the velocity response time τ_p, a dimensionless drag coefficient C_D is defined, such that

$$\tau_p = \frac{4\rho_p d_p^2}{3\mu C_D \text{Re}_r} \tag{5-4}$$

Comparing Equation 5-4 with Equation 5-3, the Stokes drag law can be stated in an alternative way, by defining the drag coefficient as

$$C_D = \frac{24}{\text{Re}_r}$$

As will be shown in the following subsections, the drag coefficient in most other drag laws shows the same asymptotic behavior as the Stokes drag law for extremely small relative Reynolds numbers.

Schiller-Naumann

When **Schiller-Naumann** is selected from the **Drag law** list, the drag coefficient becomes

$$C_D = \frac{24}{Re_r}(1 + 0.15 Re_r^{0.687})$$

The Schiller-Naumann drag law is applicable over a moderate range of relative Reynolds numbers, $Re_r < 800$.

Haider-Levenspiel

When **Haider-Levenspiel** is selected from the **Drag law** list, the drag coefficient is given by

$$C_D = \frac{24}{Re_r}\left(1 + A(S_p)Re_r^{B(S_p)}\right) + \frac{C(S_p)}{1 + D(S_p)/Re_r}$$

where A, B, C, and D (all dimensionless) 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 nonspherical particle

$$0 < S_p = \frac{A_{sphere}}{A_{particle}} \leq 1$$

The correlation coefficients are given by

$$
\begin{aligned}
A(S_p) &= \exp(2.3288\text{-}6.4581S_p + 2.4486S_p^2) \\
B(S_p) &= 0.0964 + 0.5565S_p \\
C(S_p) &= \exp(4.905\text{-}13.8944S_p + 18.4222S_p^2 - 10.2599S_p^3) \\
D(S_p) &= \exp(1.4681 + 12.2584S_p - 20.7322S_p^2 + 15.8855S_p^3)
\end{aligned}
$$

The diameter used in the Reynolds number is that of the volume equivalent sphere.

Oseen Correction

When **Oseen correction** is selected from the **Drag law** list, the drag coefficient is given by

$$C_D = \frac{24}{Re_r}\left(1 + \frac{3}{16}Re_r\right)$$

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 drag coefficient is defined as

$$C_D = \frac{8}{\text{Re}_r}\left(\frac{2 + 3\kappa}{1 + \kappa}\right) \qquad \kappa \equiv \frac{\mu_p}{\mu}$$

where μ_p (SI unit: Pa·s) is the dynamic viscosity of the droplet or gas bubble. For arbitrarily large values of μ_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 surface-active 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.

Standard Drag Correlations

The option **Standard drag correlations** defines the drag coefficient as a piecewise function of the relative Reynolds number:

RANGE	CORRELATION
$\text{Re}_r \leq 0.01$	$C_D = \frac{24}{\text{Re}_r}\left(1 + \frac{3}{16}\text{Re}_r\right)$
$0.01 < \text{Re}_r \leq 20$	$C_D = \frac{24}{\text{Re}_r}(1 + 0.1315\text{Re}_r^{0.82 - 0.05w})$
$20 < \text{Re}_r \leq 260$	$C_D = \frac{24}{\text{Re}_r}(1 + 0.1935\text{Re}_r^{0.6305})$
$260 < \text{Re}_r \leq 1500$	$\log C_D = 1.6435 - 1.1242w + 0.1558w^2$
$1500 < \text{Re}_r \leq 1.2 \times 10^4$	$\log C_D = -2.4571 + 2.5558w - 0.9295w^2 + 0.1049w^3$
$1.2 \times 10^4 < \text{Re}_r \leq 4.4 \times 10^4$	$\log C_D = -1.9181 + 0.6370w - 0.0636w^2$
$4.4 \times 10^4 < \text{Re}_r \leq 3.38 \times 10^5$	$\log C_D = -4.3390 + 1.5809w - 0.1546w^2$
$3.38 \times 10^5 < \text{Re}_r \leq 4 \times 10^5$	$C_D = 29.78 - 5.3w$

RANGE	CORRELATION
$4 \times 10^5 < \mathrm{Re_r} \leq 1 \times 10^6$	$C_D = 0.1w - 0.49$
$1 \times 10^6 < \mathrm{Re_r}$	$C_D = 0.19 - \dfrac{8 \times 10^4}{\mathrm{Re_r}}$

where $w = \log \mathrm{Re_r}$ 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.

At lower relative Reynolds numbers, this correlation agrees with the **Oseen correction**.

The drag coefficient as a function of the relative Reynolds number is shown in Figure 5-1.
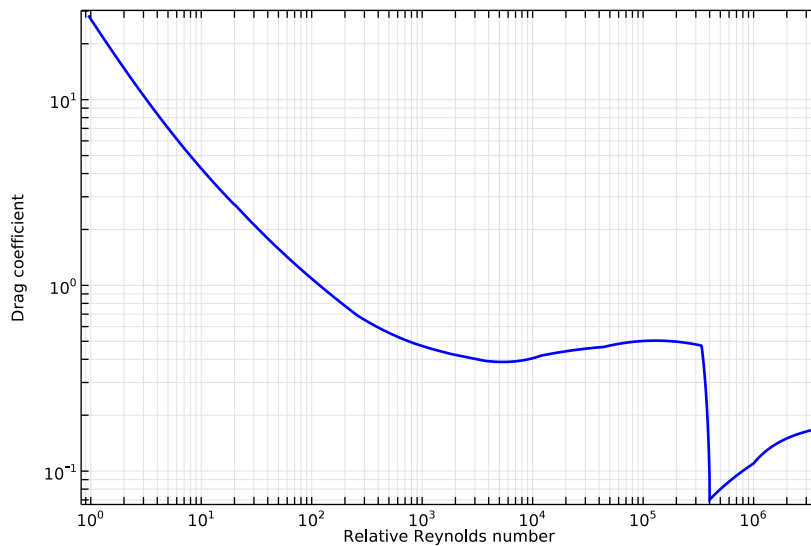
Figure 5-1: The piecewise function used to compute the drag coefficient when Standard Drag Correlations are used.

WALL CORRECTIONS

When the **Include wall corrections** check box is selected in the settings for the **Drag Force** node, the drag force is given by Ref. 5 as

$$\mathbf{F}_{\text{WC}} = M\mathbf{F}_{\text{D}}$$

$$M = \left(1 + \cfrac{1}{1 - \cfrac{9}{16}\alpha + \cfrac{1}{8}\alpha^3 - \cfrac{45}{256}\alpha^4 - \cfrac{1}{16}\alpha^5}\right)(I - P(\mathbf{n})) + \cfrac{1}{1 - \cfrac{9}{8}\alpha + \cfrac{1}{2}\alpha^3}P(\mathbf{n}) \quad (5\text{-}5)$$

$$\alpha = \frac{r_{\text{p}}}{l_{\text{W}}}$$

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), and
- l_{W} is the distance from the center of the particle to the nearest wall (SI unit: m).

The first term in Equation 5-5 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:

$$\mathbf{F}_{\text{g}} = m_{\text{p}}\mathbf{g}\frac{(\rho_{\text{p}} - \rho)}{\rho_{\text{p}}}$$

where

- ρ (SI unit: kg/m^3) is the density of the surrounding fluid,
- ρ_{p} (SI unit: kg/m^3) is the density of the surrounding fluid, and
- $\mathbf{g}$ (SI unit: m/s^2) is the gravity vector. At sea level its magnitude is approximately 9.80665 m/s^2.

Because the fluid density appears in this expression, buoyancy is accounted for.

Drag Force in a Rarefied Flow

The drag laws defined in the previous section are valid when the liquid or gas surrounding each particle can be treated as a continuum. The assumption of

continuum flow in the vicinity of a particle breaks down if the density of the fluid is extremely low, the particles are extremely small, or some combination of these factors.

If the particles are very small or the gas is extremely rarefied, consider selecting the **Include rarefaction effects** check box in the settings for the Particle Tracing for Fluid Flow interface to include some of the corrections described in this section.

DEFINING THE KNUDSEN NUMBER AND MEAN FREE PATH

The assumption of continuum flow surrounding the particle is valid when the particle Knudsen number is very small. The Knudsen number $\mathbf{Kn}$ is a dimensionless number that relates the particle size to the mean free path of molecules in the surroundings,

$$\mathbf{Kn} = \frac{\lambda}{d_p}$$

where d_p (SI unit: m) is the particle diameter and λ (SI unit: m) is the mean free path of molecules in the surrounding fluid.

⚠️ Use extra caution when interpreting expressions in terms of the Knudsen number from scientific literature. In this manual, the Knudsen number is defined in terms of particle diameter d_p, while in many references the particle radius r_p is used instead.

For the purposes of computing the particle Knudsen number, the mean free path of molecules in the surrounding fluid is defined as

$$\lambda = \sqrt{\frac{\pi}{2p\rho}}\mu \tag{5-6}$$

where

- p (SI unit: Pa) is the gas pressure,
- ρ (SI unit: kg/m^3) is the gas density, and
- μ is the gas dynamic viscosity (SI unit: Pa·s).

This is very similar to the expression given in Ref. 6, albeit with a slight simplification. Jennings (Ref. 6) defines the mean free path as

$$\lambda = \sqrt{\frac{\pi u}{8\mu}}\frac{1}{\sqrt{\rho p}} \tag{5-7}$$

Comparing Equation 5-6 to Equation 5-7, it is clear that the COMSOL implementation uses $u = 0.5$. Jennings instead suggests $u = 0.4987445$. Allen and Raabe (Ref. 7) suggest $u = 0.498$ when treating the molecules as hard elastic spheres. You can multiply the mean free path by a user-defined coefficient by selecting **User-defined correction** from the **Mean free path calculation** list in the settings for the **Drag Force** node.

APPLYING A SLIP CORRECTION TO THE DRAG COEFFICIENT

When the Knudsen number is extremely close to zero, $\text{Kn} \ll 1$, the surrounding fluid may be treated as a continuum flow; a correction factor is not necessary. However, if the Knudsen number is significantly greater than zero, rarefaction effects should be taken into account. In that case, select the **Include rarefaction effects** check box in the physics interface **Particle Release and Propagation** section. The drag force C_D (dimensionless) is then defined as the drag force for continuum flow $C_{D,C}$ (dimensionless), divided by a slip correction factor S (dimensionless):

$$C_D = \frac{C_{D,C}}{S}$$

The definition of the slip correction factor S is given for each available rarefaction model in Table 5-1 below. In the expressions for the **Basset**, **Epstein**, and **Phillips** models, the factor 2Kn is explicitly given as a reminder that these models were originally defined for a Knudsen number in terms of particle radius, not diameter. Additional details about each model are given in the ensuing subsections.

TABLE 5-1: DRAG FORCE CORRECTION FACTORS

MODEL NAME	EXPRESSION
CMD	$S = 1 + \text{Kn}\left(C_1 + C_2 \exp\left(-\frac{C_3}{\text{Kn}}\right)\right)$
Basset	$S = \dfrac{1}{1 - (2 - \sigma_R)\dfrac{(2\text{Kn})}{\sigma_R}}$ Only valid for small Kn.
Epstein	$S = \dfrac{18(2\text{Kn})}{8 + \pi\sigma}$ Asymptotic solution for large Kn.

TABLE 5-1: DRAG FORCE CORRECTION FACTORS

MODEL NAME	EXPRESSION
Phillips	$$S = \frac{15 + 12c_1(2\text{Kn}) + 9(c_1^2 + 1)(2\text{Kn})^2 + 18c_2(c_1^2 + 2)(2\text{Kn})^3}{15 - 3c_1(2\text{Kn}) + c_2(8 + \pi\sigma_R)(c_1^2 + 2)(2\text{Kn})^2}$$ $$c_1 = \frac{2 - \sigma_R}{\sigma_R} \qquad c_2 = \frac{1}{2 - \sigma_R}$$
User defined	Create user-defined expression for S.
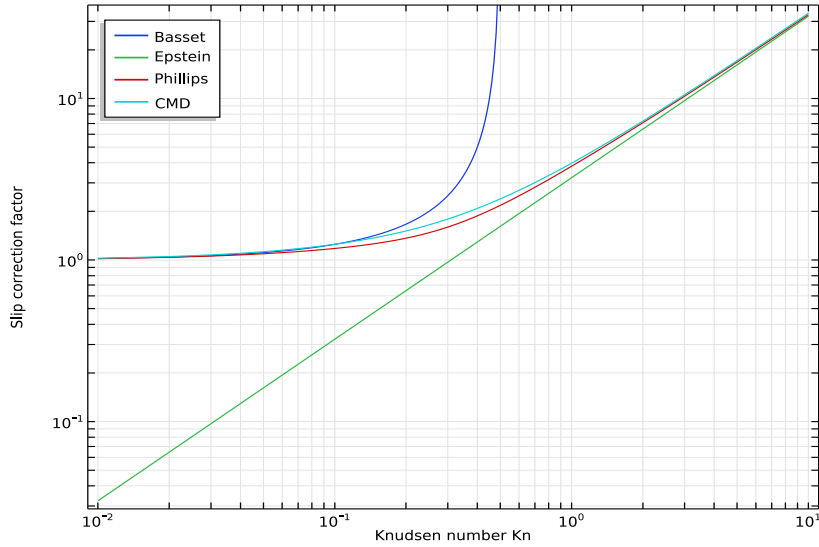
Figure 5-2: Comparison of the available slip correct factor definitions for drag force in a rarefied gas. It is clear that the Epstein model is only applicable at high Kn whereas the Basset model is only applicable at low Kn.

CUNNINGHAM-MILLIKAN-DAVIES (CMD) MODEL

The **Cunningham-Millikan-Davies** model, or CMD model, uses three dimensionless, user-defined coefficients to define the correction factor:

$$S = 1 + \text{Kn}\left(C_1 + C_2\exp\left(-\frac{C_3}{\text{Kn}}\right)\right)$$

The default values of the three coefficients are $C_1 = 2.514$, $C_2 = 0.8$, and $C_3 = 0.55$. Allen and Raabe (Ref. 7) list the coefficients given by various authors.

Note that in some references (7, 9), the Knudsen number is defined by using the particle radius as the characteristic length scale, not the diameter. When importing coefficients from such a reference, multiply the coefficients C_1 and C_2 by 2, and divide the coefficient C_3 by 2.

For example, Millikan (Ref. 9) suggested values of 0.864, 0.290, and 1.25. To use Millikan's fitting parameters in COMSOL, you should instead enter the values 1.728, 0.580, and 0.625. Even with this adjustment, the fitting parameters used by Millikan differ significantly from those used by more recent authors (Ref. 7) in part because Millikan used a different coefficient to compute the mean free path of air.

EPSTEIN MODEL

Epstein (Ref. 10) analytically derived the net force on a spherical particle in a rarefied gas in the limit of extremely high Knudsen number, so that the mean free path of molecules is much greater than the particle diameter:

$$F = \delta \frac{4\pi}{3} Nm\bar{c}a^2 V \tag{5-8}$$

where

- N (SI unit: $1/m^3$) is the number density of molecules in the gas,
- m (SI unit: kg) is the molecular mass of the gas,
- $\bar{c}$ (SI unit: m/s) is the average speed of the molecules in the gas,
- a (SI unit: m) is the particle radius, and
- V (SI unit: m/s) is the norm of the average velocity of the gas relative to the particle.

The dimensionless parameter δ has different values depending on the type of reflection the gas molecules experience on the particle surface. For specular reflection, $\delta = 1$.

For diffuse reflection, $\delta = 1 + \pi/8$. The particle is assumed to be a perfect thermal conductor. Furthermore, this type of diffuse reflection is called "diffuse reflection with accommodation", meaning that the reflected particle velocity is sampled from a Maxwell distribution based on the effective temperature of the particle. That is, both the reflected particle speed and direction are sampled randomly.

Epstein substituted into the previous expression the relationship

$$\mu = \phi Nm\bar{c}\lambda \tag{5-9}$$

where λ is the mean free path of molecules and ϕ is a dimensionless coefficient. Epstein and Millikan (Refs. 8, 9, and 10) used $\phi = 0.3502$, while subsequent authors

(Refs. 6 and 7) used values of approximately $\phi = 0.5$. Using the more up-to-date value and substituting Equation 5-9 into Equation 5-8 yields

$$F = \delta \frac{8}{18} \frac{a}{\lambda} 6\pi a \mu V$$

The last part of this expression can be recognized as the Stokes drag force in a continuum flow, so that

$$F = \frac{F_{\text{Stokes}}}{S} \qquad S = \frac{18\lambda}{8\delta a}$$

Let σ (dimensionless) be the fraction of molecules that undergo diffuse reflection with accommodation at the particle surface, while the remainder undergo specular reflection. Many authors (Refs. 7, 8, 9, and 10) report values of approximately $\sigma = 0.9$. Then

$$S = \frac{18}{8[(1 + \pi/8)\sigma + (1-\sigma)]} \frac{\lambda}{a}$$

Simplification yields

$$S = \frac{18}{8 + \pi\sigma} \frac{\lambda}{a}$$

Finally, note that the COMSOL implementation always defines the Knudsen number in terms of the particle diameter, and thus $\lambda/a = 2\text{Kn}$. The final expression for the drag correction factor for the **Epstein** model is therefore

$$S = \frac{18(2\text{Kn})}{8 + \pi\sigma}$$

BASSET MODEL

The **Basset** model is a first-order correction for relatively small Knudsen numbers; that is, terms of order λ/a are retained whereas terms of order $(\lambda/a)^2$ and higher are neglected. Under this assumption, Epstein (Ref. 10) provides a simplified version of Basset's formula for drag force with a nonzero slip factor,

$$F = \frac{6\pi\mu a V}{1 + \dfrac{\mu}{\beta a}} \tag{5-10}$$

The numerator is simply the expression for Stokes drag in a continuum flow. In the denominator, a (SI unit: m) is the particle radius, and the radio μ/β (dimensionless) is the slip factor, given as

$$\frac{\mu}{\beta} = 0.7004\left(\frac{2}{\sigma_R} - 1\right)\lambda \qquad (5\text{-}11)$$

where, similar to the **Epstein** model, σ_R is the fraction of gas molecules that undergo diffuse reflection with accommodation, and the remaining fraction are assumed to undergo specular reflection.

Substituting Equation 5-11 into Equation 5-10, and noting that $\lambda/a = 2\text{Kn}$, yields

$$F = \frac{6\pi\mu a V}{1 + 0.7004\left(\dfrac{2}{\sigma_R} - 1\right)(2\text{Kn})}$$

Finally, in Ref. 10 Epstein used a coefficient of 0.3502 when defining the mean free path, whereas in the COMSOL implementation this coefficient is 0.5, nearly equal to values given in more recent publications (Refs. 6 and 7). To compensate for this change, the factor of 0.7004 may be set to unity in the previous equation, yielding the equation for the slip correction factor implemented in COMSOL:

$$S = \frac{1}{1 - (2 - \sigma_R)\dfrac{(2\text{Kn})}{\sigma_R}}$$

where use has been made of the mathematical relation

$$\frac{1}{1 - x} = 1 + x + O(x^2)$$

in which terms of order x^2 can be dropped for x much smaller than unity.

PHILLIPS MODEL

If the **Phillips** model (Ref. 11) 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:

$$c_1 = \frac{2 - \sigma_R}{\sigma_R} \qquad c_2 = \frac{1}{2 - \sigma_R}$$

This is a moment solution to the Boltzmann equation that gives the same asymptotic behavior as the **Basset** model at low Kn and the **Epstein** model at high Kn.

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. 12): 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: m^2/s^2). When solving for the particle trajectories, the drag force is computed by first adding a random velocity perturbation 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 ε (SI unit: m^2/s^3). Therefore, the accompanying fluid flow interface should use one of the following turbulence models, for which these variables are defined:

- k-ε
- k-ω
- SST
- Low Reynolds number k-ε

> 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. 12,15,16), 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 ith 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 Weiner process with independent components, which are uncorrelated Gaussian random numbers with zero mean and variance dt (Ref. 15). A more in-depth discussion of the relevance of Weiner processes in turbulent dispersion is given in Ref. 17.

In homogeneous turbulence, the velocity perturbations du_i are computed by solving a classical Langevin equation (Refs. 15 and 16):

$$d\mathbf{u}_f = -\frac{\mathbf{u}_f}{\tau_L}dt + \sigma\sqrt{\frac{2}{\tau_L}}d\xi$$

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,

$$\sigma = \sqrt{\overline{u_{f,1}^2}} = \sqrt{\overline{u_{f,2}^2}} = \sqrt{\overline{u_{f,3}^2}} = \sqrt{\frac{2k}{3}}$$

In general, the Lagrangian integral time scale τ_L (SI unit: s) is given by (Ref. 12)

$$\tau_{L,i} = \int_0^\infty \frac{\overline{u_{f,i}(t)u_{f,i}(t+s)}}{u_{f,i}(t)^2}ds$$

Essentially, the Lagrangian time scale characterizes the time interval size over which successively sampled random numbers should become uncorrelated. Typically τ_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. 12) and as low as 1/7 (Ref. 16).

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 nonphysical 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. 16):

$$d\mathbf{u'}_f = -\frac{\mathbf{u'}_f}{\tau_L}dt + \sqrt{\frac{2}{\tau_L}}d\xi + \frac{1}{3\sigma}\frac{\partial k}{\partial \mathbf{x}}\frac{dt}{1+St}$$

The particle Stokes number St (dimensionless) is defined as

$$St = \frac{\tau_p}{\tau_L}$$

where the particle relaxation time scale τ_p (SI unit: s) or particle velocity response time depends on the drag law being used (see the The Stokes Drag Law section) and on rarefaction effects applied to the drag force, if any (see the Drag Force in a Rarefied Flow 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, wall normal, and spanwise 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 streamwise and wall normal directions. Thus the streamwise, wall normal, and spanwise 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,

$$
y^+ = \frac{x_2 u^*}{\nu}
$$

x_2 (SI unit: m) is the distance to the nearest wall, ν (SI unit: m^2/s) is the fluid kinematic 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, wall normal, and spanwise directions, respectively, are

$$
d\left(\frac{u_1}{\sigma_1}\right) = -\left(\frac{u_1}{\sigma_1}\right)\frac{dt}{\tau_1} + \sqrt{\frac{2}{\tau_1}}\,d\xi_1 + \frac{\partial(\overline{u_1 u_2/\sigma_1})}{\partial x_2}\frac{dt}{1+\mathrm{Stk}}
$$

$$
d\left(\frac{u_2}{\sigma_2}\right) = -\left(\frac{u_2}{\sigma_2}\right)\frac{dt}{\tau_2} + \sqrt{\frac{2}{\tau_2}}\,d\xi_2 + \frac{\partial\sigma_2}{\partial x_2}\frac{dt}{1+\mathrm{Stk}}
$$

$$
d\left(\frac{u_3}{\sigma_3}\right) = -\left(\frac{u_3}{\sigma_3}\right)\frac{dt}{\tau_3} + \sqrt{\frac{2}{\tau_3}}\,d\xi_3
$$

where the rms velocity perturbations (Ref. 16) are

$$\sigma_1^+ = \frac{\sigma_1}{u^*} = \frac{0.40y^+}{1 + 0.0239(y^+)^{1.496}}$$

$$\sigma_2^+ = \frac{\sigma_2}{u^*} = \frac{0.0116(y^+)^2}{1 + 0.203y^+ + 0.00140(y^+)^{2.421}}$$

$$\sigma_3^+ = \frac{\sigma_3}{u^*} = \frac{0.19y^+}{1 + 0.0361(y^+)^{1.322}}$$

The Lagrangian time scale τ_L (SI unit: s) is a piecewise function of the wall distance,

$$\tau_L = \begin{cases} \dfrac{10\rho\mu}{u^*} & y^+ \leq 5 \\[2ex] \dfrac{10\rho\mu}{(u^*)^2}(7.122 + 0.5731y^+ - 0.00129(y^+)^2) & y^+ > 5 \end{cases}$$

The dimensionless velocity perturbations are plotted in the anisotropic region in Figure 5-4. The Lagrangian time scale is plotted in Figure 5-5.

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.
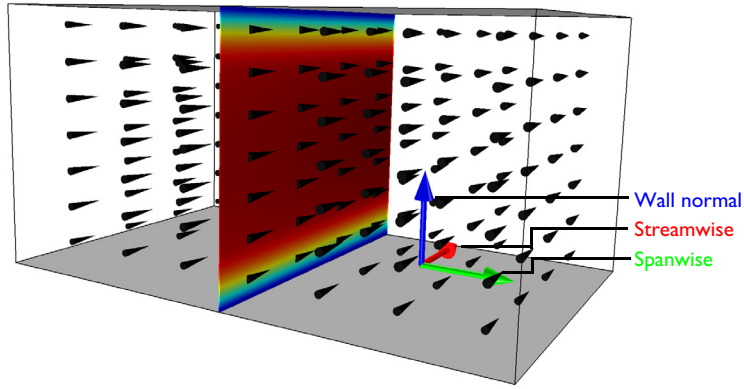
Figure 5-3: Streamwise, wall normal, and spanwise directions in a wall-bounded turbulent flow.

Figure 5-4: Dimensionless rms velocity perturbations in the streamwise, wall normal, and spanwise directions for modeling anisotropic turbulence.

Figure 5-5: Lagrangian time scale as a function of wall distance for modeling anisotropic turbulence.

DISCRETE RANDOM WALK MODEL

In the **Discrete random walk** model (Refs. [12], [20], and [21]), the components of the velocity perturbation are given instantaneous values
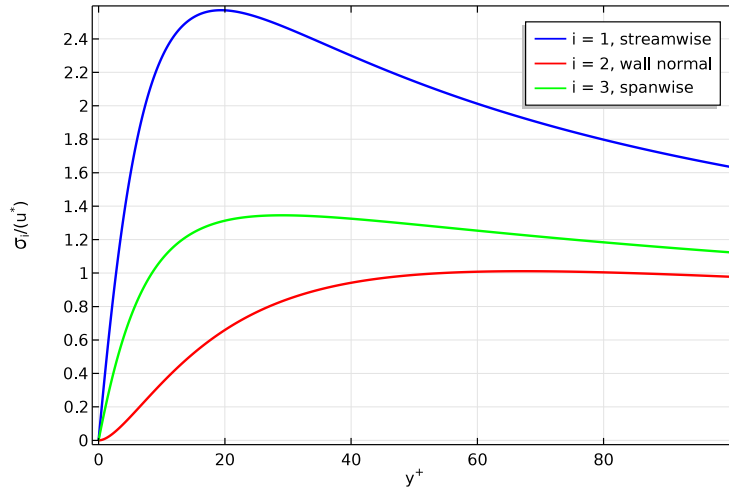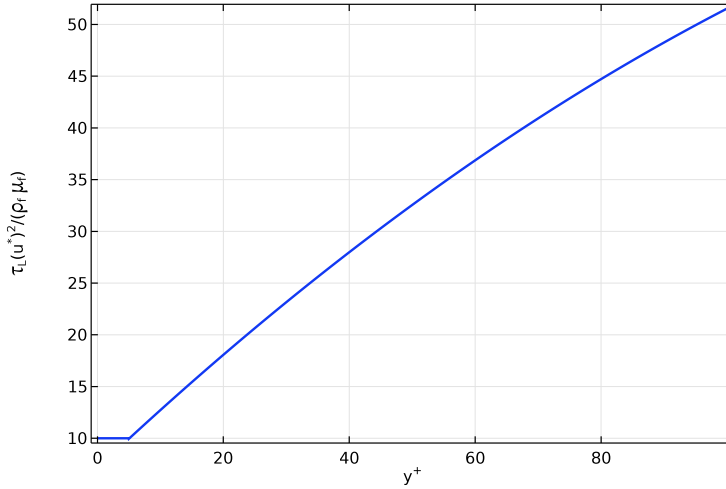
$$\mathbf{u}' = \mathbf{u} + \Delta\mathbf{u}$$

$$\Delta\mathbf{u} = \zeta\sqrt{\frac{2k}{3}}$$

where ζ (dimensionless) is a vector of uncorrelated Gaussian random numbers with unit variance (Ref. [21]).

Random Number Seeding for Velocity Perturbations

Ostensibly, the velocity perturbation used in the discrete random walk model resembles the random force contributed by the Brownian Force,

$$\mathbf{F}_{\text{ext}} = \zeta\sqrt{\frac{12\pi k_{\text{B}}\mu T r_{\text{p}}}{\Delta t}}$$

However, there is a key difference in the random number sampling used to model the molecular diffusion of the **Brownian Force**, as opposed to the background velocity perturbation used to account for turbulent dispersion. The expression for the Brownian force includes explicit inverse-square-root dependence on the size of the time step taken by the solver. Thus, it is possible to model Brownian motion using arbitrarily small time steps and resampling the vector ζ at each step, since the reduction in the denominator compensates for the fact that the random force has less time to act on the particle and resulting in statistical convergence of the particle diffusion rate.

However, the expression for the velocity perturbation used in the discrete random walk model for turbulent dispersion has no such dependence on the time step size. If ζ is naively sampled at each time step, then as the time step size is made arbitrarily small, the effect of the turbulent kinetic energy on the particle motion will eventually become negligibly small.

To ensure that the amount of turbulent diffusion is physically meaningful, it is important that the random vector ζ is only uniquely seeded at discrete time intervals. The length of the time interval is equal to the finite (not infinitesimally small) duration of time that the particle would spend, on average, interacting with a single eddy in the flow.

In the eddy interaction model of Gosman and Ioannides (Ref. 18), the time for which the random unit vector remains in a fixed direction, here called the eddy interaction time and denoted τ_i, is taken as the lesser of two different time scales,

$$\tau_i = \min(\tau_e, \tau_c)$$

The eddy lifetime τ_e (SI unit: s) is the approximate duration of time between the formation and destruction of individual eddies in the flow,

$$\tau_e = \frac{2C_L k}{\varepsilon}$$

where C_L is a dimensionless constant.

The eddy crossing time τ_c (SI unit: s) is the approximate time it takes for a particle's inertia to carry it across an eddy.

$$\tau_c = -\tau_p \log\left(1 - \frac{l_e}{\tau_p |\mathbf{u} - \mathbf{v}|}\right)$$

where τ_p is again the particle velocity response time, originally introduced in Equation 5-2 for the drag force. For Stokes drag, recall that

$$\tau_p = \frac{\rho_p d_p^2}{18\mu}$$

The dissipation length scale l_e (SI unit: m) is

$$l_e = \frac{C_L^{1/2} k^{3/2}}{\varepsilon}$$

The eddy crossing time does not have any real value if

$$\frac{l_e}{\tau |\mathbf{u} - \mathbf{v}|} > 1$$

when the physical interpretation is that the particle doesn't have sufficient inertia to escape from the eddy. It will instead continue to interact with the same eddy until the eddy dissipates, and hence $\tau_i = \tau_e$.

In order to define a random number seed that only changes at time intervals greater than the eddy interaction time τ_i, the **Discrete random walk** model defines an auxiliary dependent variable N (dimensionless) for the number of eddies encountered by a

particle in the flow. The variable N is solved computed along the particle's trajectory by integrating the first-order equation

$$\frac{dN}{dt} = \frac{1}{\tau_i}$$

The expression `floor(N)` is then used instead of the instantaneous time t as an argument in the random number seed used to compute the Gaussian random numbers ζ_i. An expression such as `randomnormal(floor(N))` only takes on a new value for each integer value that N exceeds.

⚠️ To ensure statistical convergence of the **Discrete Random Walk** model, the time steps taken by the solver should be significantly smaller than the average eddy lifetime, usually by an order of magnitude or more.

Virtual Mass Force

The virtual mass force is an optional term that can be defined by the Drag Force node by clicking the **Include virtual mass and pressure gradient forces** check box.

While the drag force is a function of the fluid viscosity, the virtual mass term is a reaction force exerted on a moving particle by the surrounding fluid, as fluid accelerates to occupy the empty space the particle leaves behind. In other words, the virtual mass force depends on the inertia of the surrounding fluid, not its viscosity.

For a particle of velocity $\mathbf{v}$ (SI unit: m/s) in a fluid of velocity $\mathbf{u}$ (SI unit: m/s), the virtual mass term $\mathbf{F}_{vm}$ (SI unit: N) is (Ref. 19)

$$\mathbf{F}_{vm} = \frac{1}{2} m_f \frac{d(\mathbf{u} - \mathbf{v})}{dt} \tag{5-12}$$

where m_f (SI unit: kg) is the mass of the fluid displaced by the particle volume,

$$m_f = \frac{1}{6} \pi d_p^3 \rho$$

d_p (SI unit: m) is the particle diameter, and ρ (SI unit: kg/m^3) is the density of the fluid at the particle's position.

The derivative term in Equation 5-12 is a total derivative or material derivative. Therefore it considers not only the time dependence of the velocity at a fixed point,

but also the motion of the particle relative to the fluid. For an arbitrary scalar field f, the material derivative is defined as

$$\frac{\mathrm{d}f}{\mathrm{d}t} \equiv \frac{\partial f}{\partial t} + \nabla f \cdot \mathbf{v}$$

Alternatively, for an arbitrary vector field $\mathbf{f}$, the material derivative is

$$\frac{\mathrm{d}\mathbf{f}}{\mathrm{d}t} \equiv \frac{\partial \mathbf{f}}{\partial t} + \nabla \mathbf{f} \cdot \mathbf{v}$$

where $\nabla \mathbf{f}$ becomes a rank 2 tensor. In either case, the dot product is taken with the particle velocity $\mathbf{v}$ instead of the fluid velocity $\mathbf{u}$, following Ref. 19.

Using the above definition of the material derivative, Equation 5-12 becomes

$$\mathbf{F}_{\mathrm{vm}} = \frac{1}{2}m_{\mathrm{f}}\left(\frac{\partial(\mathbf{u}-\mathbf{v})}{\partial t} + \nabla(\mathbf{u}-\mathbf{v}) \cdot \mathbf{v}\right)$$

Since $\mathbf{v}$ is the velocity of a discrete particle, not a field variable, its gradient $\nabla \mathbf{f}$ vanishes, leaving only

$$\mathbf{F}_{\mathrm{vm}} = \frac{1}{2}m_{\mathrm{f}}\left(\frac{\partial \mathbf{u}}{\partial t} - \frac{\partial \mathbf{v}}{\partial t} + \nabla \mathbf{u} \cdot \mathbf{v}\right)$$

The term involving the time derivative of particle velocity $\mathbf{v}$ can be moved to the left-hand side of the equation of motion, by manipulating the equation of motion of a particle as follows:

$$m_{\mathrm{p}}\frac{\partial \mathbf{v}}{\partial t} = \mathbf{F}_{\mathrm{other}} + \mathbf{F}_{\mathrm{vm}}$$

$$m_{\mathrm{p}}\frac{\partial \mathbf{v}}{\partial t} = \mathbf{F}_{\mathrm{other}} + \frac{1}{2}m_{\mathrm{f}}\left(\frac{\partial \mathbf{u}}{\partial t} - \frac{\partial \mathbf{v}}{\partial t} + \nabla \mathbf{u} \cdot \mathbf{v}\right)$$

$$\left(m_{\mathrm{p}} + \frac{1}{2}m_{\mathrm{f}}\right)\frac{\partial \mathbf{v}}{\partial t} = \mathbf{F}_{\mathrm{other}} + \frac{1}{2}m_{\mathrm{f}}\left(\frac{\partial \mathbf{u}}{\partial t} + \nabla \mathbf{u} \cdot \mathbf{v}\right)$$

$$m_{\mathrm{v}}\frac{\partial \mathbf{v}}{\partial t} = \mathbf{F}_{\mathrm{other}} + \frac{1}{2}m_{\mathrm{f}}\left(\frac{\partial \mathbf{u}}{\partial t} + \nabla \mathbf{u} \cdot \mathbf{v}\right)$$

where m_{v} (SI unit: kg) is the virtual mass,

$$m_{\mathrm{v}} = m_{\mathrm{p}} + \frac{1}{2}m_{\mathrm{f}}$$

To summarize, when including the virtual mass force, the particle mass m_{p} on the left-hand side of the equation of motion is replaced with the virtual mass m_{v}. The

contribution of the virtual mass force to the right-hand side then contains only the material derivative of the fluid velocity **u**,

$$\mathbf{F}_{mv, rhs} \; = \; \frac{1}{2}m_f\left(\frac{\partial \mathbf{u}}{\partial t} + \nabla \mathbf{u} \cdot \mathbf{v}\right)$$

(5-13)

If the virtual mass force is included in a model and the components of the virtual mass force are evaluated during results processing (for example a **Particle Evaluation** with the expression `fpt.df1.Fvmx`), the expression being evaluated will be Equation 5-13, not Equation 5-12.

Pressure Gradient Force

The pressure gradient force is an optional term that can be defined by the Drag Force node by clicking the **Include virtual mass and pressure gradient forces** check box.

For a particle of velocity **v** (SI unit: m/s) in a fluid of velocity **u** (SI unit: m/s), the pressure gradient force term $\mathbf{F}_p$ (SI unit: N) is (Ref. 19)

$$\mathbf{F}_p \; = \; m_f\frac{D\mathbf{u}}{Dt}$$

where, as in the Virtual Mass Force in the previous section, m_f (SI unit: kg) is the mass of the fluid displaced by the particle volume,

$$m_f \; = \; \frac{1}{6}\pi d_p^3\rho$$

d_p (SI unit: m) is the particle diameter, and ρ (SI unit: kg/m^3) is the density of the fluid at the particle's position.

Unlike the virtual mass force, however, the material derivative D (not d) follows the fluid velocity direction rather than the particle velocity direction. That is,

$$\frac{D\mathbf{u}}{Dt} \equiv \frac{\partial \mathbf{u}}{\partial t} + \nabla \mathbf{u} \cdot \mathbf{u}$$

or, using Einstein's notation and taking the sum over repeating indices,

$$\frac{Du_i}{Dt} \equiv \frac{\partial u_i}{\partial t} + u_j\frac{\partial u_i}{\partial x_j}$$

Particle Motion in a Shear Flow

Particles in a nonuniform 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 nonuniform 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 $\mathbf{F}_\mathrm{L}$ (SI unit: N), described in Ref. 13, is given by

$$\mathbf{F}_\mathrm{L} = 6.46 r_\mathrm{p}^2 \mathbf{L}_\mathrm{v} \sqrt{\mu \rho \frac{|\mathbf{u} - \mathbf{v}|}{|\mathbf{L}_\mathrm{v}|}}$$

$$\mathbf{L}_\mathrm{v} = (\mathbf{u} - \mathbf{v}) \times [\nabla \times (\mathbf{u} - \mathbf{v})]$$

where

- r_p is the particle radius (SI unit: m)
- μ 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. 14 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

$$\mathbf{F}_L = \rho \frac{r_p^4}{D^2} \beta(\beta G_1(s) + \gamma G_2(s))\mathbf{n}$$

$$\beta = \left| D(\mathbf{n} \cdot \nabla)\mathbf{u}_p \right|$$

$$\gamma = \left| \frac{D^2}{2}(\mathbf{n} \cdot \nabla)^2 \mathbf{u}_p \right|$$

$$\mathbf{u}_p = (\mathrm{I} - (\mathbf{n} \otimes \mathbf{n}))\mathbf{u}$$

where

- r_p is the particle radius (SI unit: m)
- μ 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. 14 (dimensionless). They are plotted in Figure 5-6 and Figure 5-7 below.
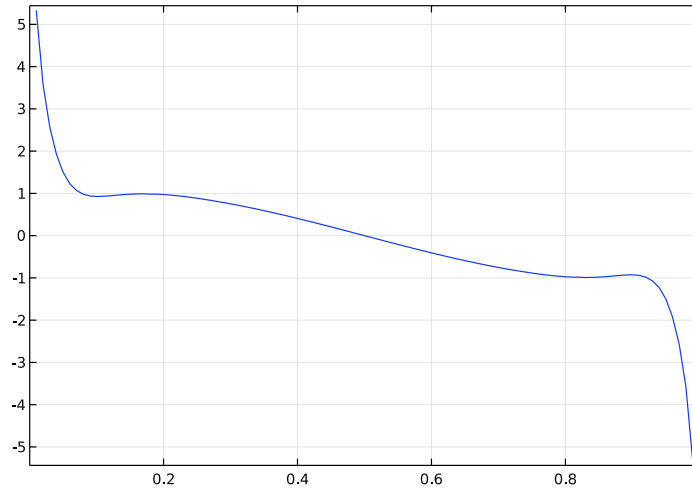
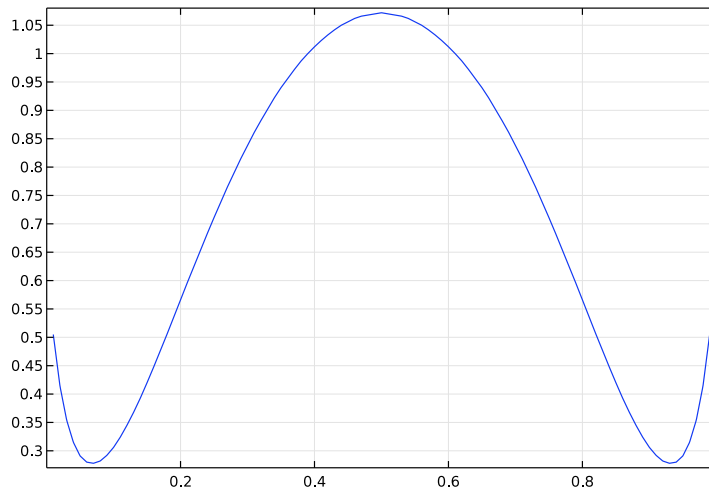*Figure 5-6: G_1 coefficient for the lift force.*

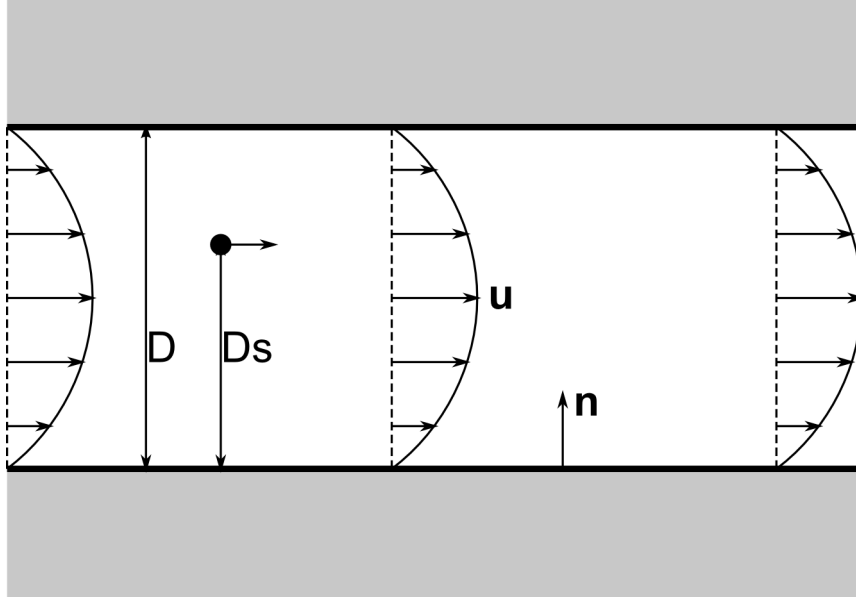*Figure 5-7: G_2 coefficient for the lift force.*

Figure 5-8: 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.

Brownian Force

The Brownian Force feature adds a Brownian force term $\mathbf{F}_b$ (SI unit: N) to the total force on the particles:

$$\mathbf{F}_b = \zeta \sqrt{\frac{12\pi k_B \mu T r_p}{\Delta t}} \tag{5-14}$$

where

- Δt (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
- μ (SI unit: Pa·s) is the fluid dynamic viscosity,
- $k_B = 1.380649 \times 10^{-23}$ J/K is the Boltzmann constant, and
- ζ (dimensionless) is a normally distributed random number with a mean of zero and unit standard deviation.

As explained in Ref. 25, independent values for ζ are chosen in all directions. A different value of ζ 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 and Magnetic Forces

The Electric Force feature exerts a force $\mathbf{F}_e$ on the particles:

$$\mathbf{F}_e = eZ\mathbf{E}$$

where $e = 1.602176634 \times 10^{-19}$ 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:

$$\mathbf{E} = -\nabla V$$

The Magnetic Force feature exerts a force $\mathbf{F}_m$ on the particles:

$$\mathbf{F}_m = eZ(\mathbf{v} \times \mathbf{B})$$

where $\mathbf{B}$ (SI unit: T) is the magnetic flux density.

MODULATION BY A PHASE FACTOR

When the electric or magnetic field is computed in the frequency domain, it becomes complex-valued. The field must be cast into a real value that depends on the angular frequency and the simulation time:

$$\mathbf{E}(t) = \text{real}(\tilde{\mathbf{E}}\exp(j(\omega t + \phi_0)))$$
$$\mathbf{B}(t) = \text{real}(\tilde{\mathbf{B}}\exp(j(\omega t + \phi_0)))$$

where $\tilde{\mathbf{E}}$ and $\tilde{\mathbf{B}}$ are the complex-valued fields solved for in the frequency domain, ω (SI unit: rad/s) is the angular frequency, ϕ_0 (SI unit: rad) is the phase angle, and t (SI unit: s) is time.

Dielectrophoretic Force

The Dielectrophoretic Force feature adds the following contribution $\mathbf{F}_{dep}$ (SI unit: N) to the total force acting on a particle. For the case of a static nonuniform electric field,

$$\mathbf{F}_{dep} = 2\pi r_p^3 \varepsilon_0 \varepsilon_r \left(\frac{\varepsilon_{r,\,p} - \varepsilon_r}{\varepsilon_{r,\,p} + 2\varepsilon_r} \right) \nabla |\mathbf{E}|^2$$

where

- r_p (SI unit: m) is the particle radius,
- $\varepsilon_0 = 8.854187817 \times 10^{-12}$ F/m is the permittivity of vacuum,
- ε_r (dimensionless) is the relative permittivity of the surrounding fluid,
- $\varepsilon_{r,p}$ (dimensionless) is the relative permittivity of the particle, and
- $\mathbf{E}$ (SI unit: V/m) is the electric field.

The electric field is assumed to change over length scales much larger than r_p.

The above expression is valid for DC fields, but if the electric field is time-harmonic, the dielectrophoretic force instead becomes

$$\mathbf{F}_{dep} = 2\pi r_p^3 \varepsilon_0 \,\text{real}(\varepsilon_r^*) \,\text{real}\left(\frac{\varepsilon_{r,\,p}^* - \varepsilon_r^*}{\varepsilon_{r,\,p}^* + 2\varepsilon_r^*} \right) \nabla |\mathbf{E}_{rms}|^2$$

where ε_r^* and $\varepsilon_{r,\,p}^*$ (both dimensionless) are the complex relative permittivity of the fluid and particle, respectively,

$$\varepsilon_r^* = \varepsilon_r - \frac{i\sigma}{\omega\varepsilon_0} \qquad \varepsilon_{r,\,p}^* = \varepsilon_{r,\,p} - \frac{i\sigma}{\omega\varepsilon_0}$$

- σ (SI unit: S/m) is the electrical conductivity of the fluid,
- σ_p (SI unit: S/m) is the electrical conductivity of the fluid, and
- $\mathbf{E}_{rms}$ denotes the root mean square electric field.

DIELECTROPHORESIS OF PARTICLES WITH THIN OUTER SHELLS

The Shell feature can be added to the Dielectrophoretic Force node to model the dielectrophoretic force on particles with thin outer layers. The volume inside the thin layers is assumed to have uniform conductivity and permittivity, but the properties of the shell can differ significantly from those of the rest of the particle. When computing the dielectrophoretic force, the complex permittivity of the particle $\varepsilon_{r,\,p}^*$ is replaced by

the equivalent complex relative permittivity $\varepsilon_{r, eq}^*$ of a homogeneous particle comprising both the shell and the interior of the particle:

$$\varepsilon_{r, eq}^* = \varepsilon_{r, s}^* \frac{\left(\dfrac{r_o}{r_i}\right)^3 + 2\left(\dfrac{\varepsilon_{r, p}^* - \varepsilon_{r, s}^*}{\varepsilon_{r, p}^* + 2\varepsilon_{r, s}^*}\right)}{\left(\dfrac{r_o}{r_i}\right)^3 - \left(\dfrac{\varepsilon_{r, p}^* - \varepsilon_{r, s}^*}{\varepsilon_{r, p}^* + 2\varepsilon_{r, s}^*}\right)}$$

where r_o and r_i (SI unit: m) are the outer and inner radii of the shell, respectively; $\varepsilon_{r, p}^*$ (dimensionless) is the complex relative permittivity of the particle, and $\varepsilon_{r, s}^*$ (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.

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:

$$\mathbf{F}_m = 2\pi r_p^3 \mu_0 \mu_r K \nabla |\mathbf{H}|^2 \tag{5-15}$$

where $\mathbf{H}$ is the magnetic field, μ_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 5-15, 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.

Acoustophoretic Radiation Force

The Acoustophoretic Radiation Force node defines a force on particles in an acoustic pressure field. For any force to be applied to the particles, the acoustic pressure field and acoustic velocity field must be solved for in a **Frequency Domain** study, prior to solving for the particle trajectories in the time domain. If the force on the particles appears to be zero, check that a **Frequency Domain** study has been selected in the **Values of variables not solved** for in the settings for the **Time Dependent** study for the particle trajectories.

The acoustophoretic radiation force is applicable for small particles in the Rayleigh limit, where the product of particle radius and wave number is much smaller than unity. In this limit, scattering theory applies and the scattering coefficients can be computed analytically for drops and solid particles. The first classical expression below is the **Ideal** model given by Gor'kov (Ref. 22). If the particle radius becomes comparable to the viscous or thermal boundary layer thicknesses of the fluid, viscous and thermal effects need to be taken into account (Ref. 23, Ref. 24). The **Viscous** and **Thermoviscous** models include these additional effects.

The acoustic radiation force $\mathbf{F}_{rad}$ is a function of the acoustic pressure field p (SI unit: Pa) and the acoustic velocity field $\mathbf{u}_{in}$ (SI unit: m/s),

$$\mathbf{F}_{rad} = -2\pi r_p^3 \left[\frac{1}{3} \kappa_s \mathrm{Re}(f_0^* p^* \nabla p) - \frac{1}{2} \rho \mathrm{Re}(f_1^* \mathbf{u}_{in}^* \cdot \nabla \mathbf{u}_{in}) \right]$$

where r_p (SI unit: m) is the particle radius and κ_s (SI unit: 1/m) is the isentropic compressibility of the fluid surrounding the particle,

$$\kappa_s = \frac{1}{\rho c^2}$$

ρ (SI unit: kg/m^3) is the fluid density and c (SI unit: m/s) is its adiabatic sound speed.

The dimensionless quantities f_0 and f_1 are respectively the monopole and dipole scattering coefficients. Their expressions change depending on whether the particles are solids or liquid, and on whether the effects of viscosity and thermal conductivity are considered in the derivation of the force.

The **Acoustophoretic Radiation Force** supports three different types of **Thermodynamic loss model**: **Ideal**, **Viscous**, and **Thermoviscous**. In addition, the model particles may be solid particles or liquid droplets. In total there are six different expressions for the monopole and dipole scattering coefficients. In the following sections, the monopole and dipole scattering coefficients will be indicated with a superscript sl if the particles are solid, or a superscript fl if the particles are liquid droplets.

IDEAL MODEL, SOLID PARTICLES

If **Solid particle** is selected from the **Particle type** list, and **Ideal** is selected from the **Thermodynamic loss model** list, then the monopole and dipole scattering coefficients are

$$f_0^{sl} = 1 - \tilde{\kappa}_s$$

$$f_1^{sl} = \frac{2(\tilde{\rho} - 1)}{2\tilde{\rho} + 1}$$

where the expressions under ~ indicate ratios of the particle and fluid properties,

$$\tilde{\kappa}_s = \frac{\kappa_{s,p}}{\kappa_s} \qquad \tilde{\rho} = \frac{\rho_p}{\rho}$$

The isentropic compressibility of the particle and fluid, respectively, are

$$\kappa_{s,p} = \frac{1}{\rho_p\left(c_{p,p}^2 - \frac{4}{3}c_{s,p}^2\right)} \qquad \kappa_s = \frac{1}{\rho c^2}$$

- ρ_p (SI unit: kg/m^3) is the particle density,
- $c_{p,p}$ (SI unit: m/s) is the pressure-wave speed of the solid particle,
- $c_{s,p}$ (SI unit: m/s) is the shear-wave speed of the solid particle,
- ρ (SI unit: kg/m^3) is the fluid density, and
- c (SI unit: m/s) is the fluid speed of sound.

IDEAL MODEL, LIQUID DROPLETS

If **Liquid droplet** is selected from the **Particle type** list, and **Ideal** is selected from the **Thermodynamic loss model** list, then the monopole and dipole scattering coefficients are

$$f_0^{fl} = 1 - \tilde{\kappa}_s$$

$$f_1^{fl} = \frac{2(\tilde{\rho} - 1)}{2\tilde{\rho} + 1}$$

where the expressions under ~ indicate ratios of the particle and fluid properties,

$$\tilde{\kappa}_s = \frac{\kappa_{s,p}}{\kappa_s} \qquad \tilde{\rho} = \frac{\rho_p}{\rho}$$

The isentropic compressibility of the particle and fluid, respectively, are

$$\kappa_{s,p} = \frac{1}{\rho_p c_p^2} \qquad \kappa_s = \frac{1}{\rho c^2}$$

- ρ_p (SI unit: kg/m^3) is the droplet density,
- c_p (SI unit: m/s) is the droplet speed of sound,
- ρ (SI unit: kg/m^3) is the fluid density, and
- c (SI unit: m/s) is the fluid speed of sound.

VISCOUS MODEL, SOLID PARTICLES

If **Solid particle** is selected from the **Particle type** list, and **Viscous** is selected from the **Thermodynamic loss model** list, then the monopole and dipole scattering coefficients are

$$f_0^{\text{sl}} = 1 - \tilde{\kappa}_s$$

$$f_1^{\text{sl}} = \frac{2(\tilde{\rho} - 1)[1 - G(x_s)]}{2\tilde{\rho} + 1 - 3G(x_s)}$$

where the expressions under ~ indicate ratios of the particle and fluid properties,

$$\tilde{\kappa}_s = \frac{\kappa_{s,p}}{\kappa_s} \qquad \tilde{\rho} = \frac{\rho_p}{\rho}$$

The isentropic compressibility of the particle and fluid, respectively, are

$$\kappa_{s,p} = \frac{1}{\rho_p\left(c_{p,p}^2 - \frac{4}{3}c_{s,p}^2\right)} \qquad \kappa_s = \frac{1}{\rho c^2}$$

- ρ_p (SI unit: kg/m^3) is the particle density,
- $c_{p,p}$ (SI unit: m/s) is the pressure-wave speed of the solid particle,
- $c_{s,p}$ (SI unit: m/s) is the shear-wave speed of the solid particle,
- ρ (SI unit: kg/m^3) is the fluid density, and
- c (SI unit: m/s) is the fluid speed of sound.

The function $G(x_s)$ is a dimensionless help function,

$$G(x_s) = \frac{3}{x_s}\left(\frac{1}{x_s} - i\right)$$

x_s is the dimensionless shear wave number of the fluid,

$$x_s = k_s r_p$$

and in turn r_p (SI unit: m) is the particle radius and k_s (SI unit: 1/m) is the shear (or viscous) wave number of the fluid,

$$k_s = \frac{1 + i}{\delta_s}$$

δ_s (SI unit: m) is the shear (or viscous) boundary layer thickness,

$$\delta_s = \sqrt{\frac{2\mu}{\rho\omega}}$$

μ (SI unit: Pa·s) is the fluid dynamic viscosity, and ω (SI unit: rad/m) is the angular frequency of the acoustic pressure field.

VISCOUS MODEL, LIQUID DROPLETS

If **Liquid droplet** is selected from the **Particle type** list, and **Viscous** is selected from the **Thermodynamic loss model** list, then the monopole and dipole scattering coefficients are

$$f_0^{fl} = 1 - \tilde{\kappa}_s$$

$$f_1^{sl} = \frac{2(\tilde{\rho} - 1)[1 + F(x_s, x_{s, p}) - G(x_s)]}{(2\tilde{\rho} + 1)[1 + F(x_s, x_{s, p})] - 3G(x_s)}$$

where the expressions under ~ indicate ratios of the particle and fluid properties,

$$\tilde{\kappa}_s = \frac{\kappa_{s, p}}{\kappa_s} \qquad \tilde{\rho} = \frac{\rho_p}{\rho}$$

The isentropic compressibility of the particle and fluid, respectively, are

$$\kappa_{s, p} = \frac{1}{\rho_p c_p^2} \qquad \kappa_s = \frac{1}{\rho c^2}$$

- ρ_p (SI unit: kg/m^3) is the droplet density,
- c_p (SI unit: m/s) is the droplet speed of sound,
- ρ (SI unit: kg/m^3) is the fluid density, and
- c (SI unit: m/s) is the fluid speed of sound.

The functions $F(x_s, x_{s,p})$ and $G(x_s)$ are dimensionless help functions,

$$F(x_s, x_{s, p}) = \frac{1 - ix_s}{2(1 - \tilde{\mu}) + \dfrac{\tilde{\mu}x_{s, p}^2(\tan x_{s, p} - x_{s, p})}{(3 - x_{s, p}^2)\tan x_{s, p} - 3x_{s, p}}}$$

$$G(x_s) = \frac{3}{x_s}\left(\frac{1}{x_s} - i\right)$$

the dimensionless dynamic viscosity ratio is

$$\tilde{\mu} = \frac{\mu_p}{\mu}$$

x_s and $x_{s,p}$ are respectively the dimensionless shear wave numbers of the surrounding fluid and the droplet,

$$x_s = k_s r_p \qquad x_{s,p} = k_{s,p} r_p$$

and in turn r_p (SI unit: m) is the particle radius, and k_s and $k_{s,p}$ (SI unit: 1/m) are respectively the shear (or viscous) wave number of the surrounding fluid and the droplet,

$$k_s = \frac{1+i}{\delta_s} \qquad k_{s,p} = \frac{1+i}{\delta_{s,p}}$$

δ_s and $\delta_{s,p}$ (SI unit: m) are respectively the shear (or viscous) boundary layer thickness of the surrounding fluid and the droplet,

$$\delta_s = \sqrt{\frac{2\mu}{\rho\omega}} \qquad \delta_{s,p} = \sqrt{\frac{2\mu_p}{\rho_p\omega}}$$

μ (SI unit: Pa·s) is the dynamic viscosity of the surrounding fluid, μ_p (SI unit: Pa·s) is the dynamic viscosity of the liquid droplet, and ω (SI unit: rad/m) is the angular frequency of the acoustic pressure field.

THERMOVISCOUS MODEL, SOLID PARTICLES

If **Solid particle** is selected from the **Particle type** list, and **Thermoviscous** is selected from the **Thermodynamic loss model** list, then the monopole and dipole scattering coefficients are

$$f_0^{sl} = \frac{1 - \tilde{\kappa}_s + 3(\gamma-1)\left[\left(1 - \frac{\tilde{\alpha}_p}{\tilde{\rho}\tilde{C}_p}\right)\left(1 - \frac{\chi_p\tilde{\alpha}_p}{\tilde{\rho}\tilde{C}_p}\right) - \frac{4}{3}\frac{\chi_p\tilde{\alpha}_p\tilde{\kappa}_s}{\tilde{C}_p}\left(\frac{c_{s,p}}{c_{p,p}}\right)^2\left(1 - \frac{\tilde{\alpha}_p}{\tilde{\rho}\tilde{C}_p\tilde{\kappa}_s}\right)\right]H}{1 + 4(\gamma-1)\frac{\chi_p\tilde{\alpha}_p^2}{\tilde{\rho}\tilde{C}_p^2}\left(\frac{c_{s,p}}{c_{p,p}}\right)^2 H}$$

$$f_1^{sl} = \frac{2(\tilde{\rho}-1)[1 - G(x_s)]}{2\tilde{\rho} + 1 - 3G(x_s)}$$

where the expressions under ~ indicate ratios of the particle and fluid properties,

$$\tilde{\kappa}_s = \frac{\kappa_{s,p}}{\kappa_s} \qquad \tilde{\rho} = \frac{\rho_p}{\rho} \qquad \tilde{\alpha}_p = \frac{\alpha_{p,p}}{\alpha_p} \qquad \tilde{C}_p = \frac{C_{p,p}}{C_p}$$

The isentropic compressibility of the particle and fluid, respectively, are

$$\kappa_{s,p} = \frac{1}{\rho_p\left(c_{p,p}^2 - \frac{4}{3}c_{s,p}^2\right)} \qquad \kappa_s = \frac{1}{\rho c^2}$$

- ρ_p (SI unit: kg/m^3) is the particle density,
- $c_{p,p}$ (SI unit: m/s) is the pressure-wave speed of the solid particle,
- $c_{s,p}$ (SI unit: m/s) is the shear-wave speed of the solid particle,
- $\alpha_{p,p}$ (SI unit: 1/K) is the isobaric coefficient of thermal expansion of the particle,
- $C_{p,p}$ (SI unit: J/(kg·K)) is the particle thermal conductivity,
- ρ (SI unit: kg/m^3) is the fluid density,
- c (SI unit: m/s) is the fluid speed of sound,
- α_p (SI unit: 1/K) is the isobaric coefficient of thermal expansion of the fluid, and
- C_p (SI unit: J/(kg·K)) is the fluid thermal conductivity.

The functions $G(x_s)$ and $H(x_{th}, x_{th,p})$ are dimensionless help functions,

$$G(x_s) = \frac{3}{x_s}\left(\frac{1}{x_s} - i\right)$$

$$H(x_{th}, x_{th,p}) = \frac{1}{x_{th}^2}\left[\frac{1}{1 - ix_{th}} - \frac{1}{\tilde{k}}\frac{\tan x_{th,p}}{\tan x_{th,p} - x_{th,p}}\right]^{-1}$$

In the thermal conductivity ratio,

$$\tilde{k} = \frac{k_p}{k}$$

both the particle thermal conductivity k_p and the fluid thermal conductivity k (SI unit: W/(m K)) are assumed isotropic, hence they are treated as scalar quantities. In addition, x_{th} is the dimensionless thermal wave number of the fluid, $x_{th,p}$ is the dimensionless thermal wave number of the particle, and x_s is the dimensionless shear wave number of the fluid,

$$x_{th} = k_{th}r_p \qquad x_{th,p} = k_{th,p}r_p \qquad x_s = k_s r_p$$

and in turn r_p (SI unit: m) is the particle radius and k_s (SI unit: 1/m) is the shear (or viscous) wave number of the fluid,

$$k_s = \frac{1+i}{\delta_s}$$

δ_s (SI unit: m) is the shear (or viscous) boundary layer thickness,

$$\delta_s = \sqrt{\frac{2\mu}{\rho\omega}}$$

μ (SI unit: Pa·s) is the fluid dynamic viscosity, and ω (SI unit: rad/m) is the angular frequency of the acoustic pressure field.

k_{th} and $k_{th,p}$ (SI unit: 1/m) are, respectively, the thermal wave numbers of the surrounding fluid and the solid particle,

$$k_{th} = \frac{1+i}{\delta_{th}}\left[1 + \frac{1}{2}(\gamma_p - 1)(\Gamma_s - \Gamma_{th})\right] \qquad k_{th,\,p} = \frac{1+i}{\delta_{th,\,p}}\frac{1}{\sqrt{1-X_p}}\left[1 + \frac{i}{8}\frac{\gamma_p^2\Gamma_{th,\,p}}{1-X_p}\right]$$

In the surrounding fluid, δ_{th} (SI unit: m) is the thermal boundary layer thickness,

$$\delta_{th} = \sqrt{\frac{2k}{\rho C_p \omega}}$$

γ (dimensionless) is the ratio of specific heats, Γ_s (dimensionless) is the viscous bulk damping factor, and Γ_{th} (dimensionless) is the thermal bulk damping factor. The bulk damping factors are defined as

$$\Gamma_s = \frac{1}{2}(1 + \beta)(k_0\delta_s)^2 \qquad \Gamma_{th} = \frac{1}{2}(k_0\delta_{th})^2 \qquad \beta = \frac{\mu_B}{\mu} + \frac{1}{3}$$

where μ_B (SI unit: Pa·s) is the fluid bulk viscosity and k_0 (SI unit: rad/m) is the lossless wave number,

$$k_0 = \frac{\omega}{c}$$

For the solid particle, $\delta_{th,p}$ (SI unit: m) is the thermal boundary layer thickness,

$$\delta_{th,\,p} = \sqrt{\frac{2k_p}{\rho_p C_{p,\,p}\omega}}$$

γ_p (dimensionless) is the ratio of specific heats, and Γ_{th} (dimensionless) is the thermal bulk damping factor,

$$\Gamma_{th,\,p} = \frac{1}{2}(k_{0,\,p}\delta_{th,\,p})^2$$

X_p is a dimensionless help variable,

$$X_p = (\gamma_p - 1)(1 - \chi_p) \qquad \chi_p = 1 - \frac{4}{3}\frac{c_{s,\,p}^2}{c_{p,\,p}^2}$$

The lossless wave number of the particle, $k_{0,p}$ (SI unit: rad/m) is based on the compressional speed of sound,

$$k_{0,\,p} = \frac{\omega}{c_{p,\,p}}$$

THERMOVISCOUS MODEL, LIQUID DROPLETS

If **Liquid droplet** is selected from the **Particle type** list, and **Thermoviscous** is selected from the **Thermodynamic loss model** list, then the monopole and dipole scattering coefficients are

$$f_0^{fl} = 1 - \tilde{\kappa}_s + 3(\gamma - 1)\left(1 - \frac{\tilde{\alpha}_p}{\tilde{\rho}\tilde{C}_p}\right)^2 H(x_{th}, x_{th,\,p})$$

$$f_1^{fl} = \frac{2(\tilde{\rho} - 1)[1 + F(x_s, x_{s,\,p}) - G(x_s)]}{(2\tilde{\rho} + 1)[1 + F(x_s, x_{s,\,p})] - 3G(x_s)}$$

where the expressions under ~ indicate ratios of the particle and fluid properties,

$$\tilde{\kappa}_s = \frac{\kappa_{s,\,p}}{\kappa_s} \qquad \tilde{\rho} = \frac{\rho_p}{\rho} \qquad \tilde{\alpha}_p = \frac{\alpha_{p,\,p}}{\alpha_p} \qquad \tilde{C}_p = \frac{C_{p,\,p}}{C_p}$$

The isentropic compressibility of the droplet and surrounding fluid, respectively, are

$$\kappa_{s,\,p} = \frac{1}{\rho_p c_p^2} \qquad \kappa_s = \frac{1}{\rho c^2}$$

- ρ_p (SI unit: kg/m^3) is the droplet density,
- c_p (SI unit: m/s) is the adiabatic sound speed of the droplet,
- $\alpha_{p,p}$ (SI unit: 1/K) is the isobaric coefficient of thermal expansion of the droplet,
- $C_{p,p}$ (SI unit: J/(kg·K)) is the droplet thermal conductivity,

- ρ (SI unit: kg/m^3) is the fluid density,
- c (SI unit: m/s) is the fluid speed of sound,
- α_p (SI unit: 1/K) is the isobaric coefficient of thermal expansion of the fluid, and
- C_p (SI unit: J/(kg·K)) is the fluid thermal conductivity.

The functions $F(x_s, x_{s,p})$, $G(x_s)$, and $H(x_{th}, x_{th,p})$ are dimensionless help functions,

$$F(x_s, x_{s,\,p}) = \cfrac{1 - ix_s}{2(1 - \tilde{\mu}) + \cfrac{\tilde{\mu} x_{s,\,p}^2 (\tan x_{s,\,p} - x_{s,\,p})}{(3 - x_{s,\,p}^2)\tan x_{s,\,p} - 3x_{s,\,p}}}$$

$$G(x_s) = \frac{3}{x_s}\left(\frac{1}{x_s} - i\right)$$

$$H(x_{th}, x_{th,\,p}) = \frac{1}{x_{th}^2}\left[\frac{1}{1 - ix_{th}} - \frac{1}{\tilde{k}}\frac{\tan x_{th,\,p}}{\tan x_{th,\,p} - x_{th,\,p}}\right]^{-1}$$

In the thermal conductivity ratio,

$$\tilde{k} = \frac{k_p}{k}$$

both the droplet thermal conductivity k_p and the surrounding fluid thermal conductivity k (SI unit: W/(m K)) are assumed isotropic, hence they are treated as scalar quantities. In addition, x_{th} is the dimensionless thermal wave number of the surrounding fluid, $x_{th,p}$ is the dimensionless thermal wave number of the droplet, x_s is the dimensionless shear wave number of the surrounding fluid, and $x_{th,p}$ is the dimensionless shear wave number of the droplet,

$$x_{th} = k_{th}r_p \qquad x_{th,\,p} = k_{th,\,p}r_p \qquad x_s = k_s r_p \qquad x_{s,\,p} = k_{s,\,p}r_p$$

and in turn r_p (SI unit: m) is the droplet radius, k_s (SI unit: 1/m) is the shear (or viscous) wave number of the fluid, and $k_{s,p}$ (SI unit: 1/m) is the shear (or viscous) wave number of the droplet,

$$k_s = \frac{1 + i}{\delta_s} \qquad k_{s,\,p} = \frac{1 + i}{\delta_{s,\,p}}$$

δ_s and $\delta_{s,p}$ (SI unit: m) are respectively the shear (or viscous) boundary layer thickness of the surrounding fluid and the droplet,

$$\delta_s = \sqrt{\frac{2\mu}{\rho\omega}} \qquad \delta_{s,\,p} = \sqrt{\frac{2\mu_p}{\rho_p\omega}}$$

μ and μ_p (SI unit: Pa·s) are respectively the dynamic viscosity of the surrounding fluid and the droplet, and ω (SI unit: rad/m) is the angular frequency of the acoustic pressure field.

k_{th} and $k_{th,p}$ (SI unit: 1/m) are, respectively, the thermal wave numbers of the surrounding fluid and the droplet,

$$k_{th} = \frac{1+i}{\delta_{th}}\left[1 + \frac{1}{2}(\gamma-1)(\Gamma_s - \Gamma_{th})\right]$$
$$k_{th,\,p} = \frac{1+i}{\delta_{th,\,p}}\left[1 + \frac{1}{2}(\gamma_p-1)(\Gamma_{s,\,p} - \Gamma_{th,\,p})\right]$$

where

- γ (dimensionless) is the ratio of specific heats of the surrounding fluid,
- γ_p (dimensionless) is the ratio of specific heats of the droplet,
- δ_{th} (SI unit: m) is the thermal boundary layer thickness of the surrounding fluid,
- $\delta t_{h,p}$ (SI unit: m) is the thermal boundary layer thickness of the droplet,
- Γ_s (dimensionless) is the shear bulk damping factor of the fluid,
- $\Gamma_{s,p}$ (dimensionless) is the shear bulk damping factor of the droplet,
- Γ_{th} (dimensionless) is the thermal bulk damping factor of the fluid, and
- Γ_{th} (dimensionless) is the thermal bulk damping factor of the droplet.

The thermal boundary layer thicknesses are defined as

$$\delta_{th} = \sqrt{\frac{2k}{\rho C_p\omega}} \qquad \delta_{th,\,p} = \sqrt{\frac{2k_p}{\rho_p C_{p,\,p}\omega}}$$

The remaining quantities are

$$\Gamma_s = \frac{1}{2}(1+\beta)(k_0\delta_s)^2 \qquad\qquad \Gamma_{th} = \frac{1}{2}(k_0\delta_{th})^2 \qquad\qquad \beta = \frac{\mu_B}{\mu} + \frac{1}{3}$$

$$\Gamma_{s,\,p} = \frac{1}{2}(1+\beta_p)(k_{0,\,p}\delta_{s,\,p})^2 \qquad \Gamma_{th,\,p} = \frac{1}{2}(k_{0,\,p}\delta_{th,\,p})^2 \qquad \beta_p = \frac{\mu_{B,\,p}}{\mu_p} + \frac{1}{3}$$

where μ_B and $\mu_{B,p}$ (SI unit: Pa·s) are respectively the bulk viscosity of the surrounding fluid and the droplet. The lossless wave numbers are defined in terms of the speed of sound,

$$k_0 = \frac{\omega}{c} \qquad k_{0,\,p} = \frac{\omega}{c_p}$$

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,
- μ (SI unit: Pa·s) is the fluid dynamic viscosity,
- ρ (SI unit: kg/m^3) 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,
- $k_B = 1.380649 \times 10^{-23}$ 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:

$$\mathbf{F}_{tp} = -\frac{6\pi d_p \mu^2}{\rho}\frac{C_s(\Lambda + C_t \mathrm{Kn})}{(1 + 3C_m \mathrm{Kn})(1 - 2\Lambda + 2C_t \mathrm{Kn})}\frac{\nabla T}{T}$$

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. 26). 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

$$R = \frac{\sqrt{\pi}}{3\mathrm{PrKn}}$$

where Pr (dimensionless) is the fluid Prandtl number,

$$\mathrm{Pr} = \frac{\mu C_p}{k}$$

and α_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 ϕ_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. 26).

R	f_{11}	f_{12}	f_{21}	f_{22}
1000	-1.324×10^{-5}	1.317×10^{-5}	1.366×10^{-9}	4.924×10^{-8}
800	-2.067×10^{-5}	2.055×10^{-5}	2.677×10^{-9}	9.601×10^{-8}
600	-3.673×10^{-5}	3.645×10^{-5}	6.379×10^{-9}	2.270×10^{-7}
400	-8.256×10^{-5}	8.159×10^{-5}	2.175×10^{-8}	7.619×10^{-7}
200	-3.291×10^{-4}	3.215×10^{-4}	1.791×10^{-7}	5.997×10^{-6}
100	-1.307×10^{-3}	1.248×10^{-3}	1.502×10^{-6}	4.643×10^{-5}
80	-2.036×10^{-3}	1.921×10^{-3}	2.987×10^{-6}	8.917×10^{-5}
60	-3.598×10^{-3}	3.331×10^{-3}	7.225×10^{-6}	2.055×10^{-4}
40	-8.004×10^{-3}	7.139×10^{-3}	2.439×10^{-5}	6.551×10^{-4}
20	-3.115×10^{-2}	2.505×10^{-2}	1.066×10^{-4}	4.417×10^{-3}
10	-0.1109	7.525×10^{-2}	-1.852×10^{-3}	2.639×10^{-2}
8	-0.1636	0.1039	-6.174×10^{-3}	4.513×10^{-2}
6	-0.2626	0.1524	-2.244×10^{-2}	8.749×10^{-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	-1.237×10^{1}	7.214
0.2	-2.717	0.9310	-2.889×10^{1}	1.562×10^{1}
0.1	-2.857	0.9678	-6.220×10^{1}	3.238×10^{1}
0.08	-2.884	0.9748	-7.885×10^{1}	4.073×10^{1}
0.06	-2.910	0.9816	-1.066×10^{2}	5.463×10^{1}
0.04	-2.835	0.9881	-1.620×10^{2}	8.239×10^{1}

R	f_{31}	f_{32}	f_{41}	f_{42}
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

R	f_{31}	f_{32}	f_{41}	f_{42}
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
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^{-2}	−2.691	0.2265
0.4	1	-5.344×10^{-2}	−2.781	0.1482
0.2	1	-2.237×10^{-2}	−2.881	6.437×10^{-2}
0.1	1	-8.413×10^{-3}	−2.937	2.470×10^{-2}
0.08	1	-5.993×10^{-3}	−2.945	1.767×10^{-2}
0.06	1	-3.793×10^{-3}	−2.961	1.123×10^{-2}
0.04	1	-1.900×10^{-3}	−2.973	5.648×10^{-3}

Note that this data is tabulated assuming that the fluid Prandtl number is 2/3.

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. 28) defined the volume removed from a surface as:

$$V = \frac{cMU^2}{4p\left(1 + \frac{mr^2}{I}\right)} f(\alpha)$$

$$f(\alpha) = \begin{cases} \cos^2\alpha & \tan\alpha > \dfrac{P}{2} \\[2ex] \dfrac{2}{P}\left[\sin(2\alpha) - \dfrac{2}{P}\sin^2\alpha\right] & \tan\alpha \leq \dfrac{P}{2} \end{cases}$$

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 (SI unit: m/s) is the magnitude of the incident particle velocity.
- p (SI unit: 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^2) is the moment of inertia of an individual particle about its center of mass. For an isotropic sphere, $I = 2mr^2/5$.
- α (SI unit: 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 BH is the Brinell hardness of the wall material (all dimensionless). The angle of incidence α 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^3/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:

$$g(a) = (\sin\alpha)^{n_1}\left[1 + \frac{Hv}{1[GPa]}(1 - \sin\alpha)\right]^{n_2}$$

$$E_{90} = \left(\frac{Hv}{1[GPa]}\right)^{k_1}\left(\frac{v}{v'}\right)^{k_2}\left(\frac{D}{D'}\right)^{k_3}$$

where v' is the reference velocity; D' is the reference diameter; the dimensionless parameters n_1, n_2, and k_2 are defined as

$$n_1 = s_1\left(\frac{Hv}{1[GPa]}\right)^{q_1}$$

$$n_2 = s_2\left(\frac{Hv}{1[GPa]}\right)^{q_2}$$

$$k_2 = 2.3\left(\frac{Hv}{1[GPa]}\right)^{0.038}$$

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:

$$E = K\left(\frac{v}{1[m/s]}\right)^{-n}F(\alpha)$$

$$F(a) = 9.370\alpha - 42.295\alpha^2 + 110.864\alpha^3 - 175.804\alpha^4$$
$$+ 170.137\alpha^5 - 98.398\alpha^6 + 31.211\alpha^7 - 4.170\alpha^8$$

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 5-2.

TABLE 5-2: DIMENSIONLESS QUANTITIES USED IN DROPLET BREAKUP MODELS

SYMBOL	NAME	EXPRESSION
We_g	Gas Weber number	$\text{We}_g = \dfrac{\rho U_{rel}^2 r_p}{\sigma_p}$
We_l	Liquid Weber number	$\text{We}_l = \dfrac{\rho_p U_{rel}^2 r_p}{\sigma_p}$
Re_l	Droplet Reynolds number	$\text{Re}_l = \dfrac{\rho_p r_p U_{rel}}{\mu_p}$
Z	Ohnesorge number	$Z = \dfrac{\sqrt{\text{We}_l}}{\text{Re}_l}$
T	Taylor number	$T = Z\sqrt{\text{We}_g}$

where

- ρ (SI unit: kg/m^3) is the gas density,
- ρ_p (SI unit: kg/m^3) is the droplet density,
- U_{rel} (SI unit: m/s) is the relative speed between the droplet and gas,
- r_p (SI unit: m) is the droplet radius,
- σ_p (SI unit: N/m) is the surface tension, and
- μ_p (SI unit: 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: m), the wavelength Λ_{KH} (SI unit: m) of the fastest-growing disturbance on the surface of the particle is

$$\Lambda_{KH} = \frac{9.02 r_0 (1 + 0.45 \sqrt{Z_l})(1 + 0.4 T^{0.7})}{(1 + 0.865 \mathrm{We}_g^{1.67})^{0.6}}$$

The corresponding growth rate Ω_{KH} (SI unit: 1/s) of the fastest-growing disturbance is

$$\Omega_{KH} = \frac{0.34 + 0.385 \mathrm{We}_g^{1.5}}{(1 + Z_l)(1 + 1.4 T^{0.6})} \sqrt{\frac{\sigma}{\rho_p r_0^3}}$$

The characteristic breakup time τ_{KH} (SI unit: s) can be derived from the wavelength and frequency of the fastest-growing disturbance,

$$\tau_{KH} = \frac{3.788 B_{KH} r_0}{\Lambda_{KH} \Omega_{KH}}$$

where B_{KH} is a dimensionless constant. The value of B_{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 τ_{KH}. The approximate size of the parent droplet over time is computed by allocating an effective Kelvin-Helmholtz radius r_{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

$$\frac{dr_{KH}}{dt} = \frac{r_{ch} - r_{KH}}{\tau_{KH}}$$

Where r_{ch} (SI unit: m) is the child droplet radius,

$$r_{ch} = \begin{cases} B_0 \Lambda_{KH} & B_0 \Lambda_{KH} \leq r_{KH} \\ \min\left[\left(\frac{3 \pi r_{KH}^2 U_{rel}}{2 \Omega_{KH}} \right)^{1/3}, \left(\frac{3 \pi r_{KH}^2 U_{rel}}{4} \right)^{1/3} \right] & B_0 \Lambda_{KH} > r_{KH} \end{cases}$$

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_{ch} such that

$$m_0 \geq m + n_{ch}r_{ch}$$

When this criterion is true, a number of child droplets equal to n_{ch} is released at the parent droplet's position. The new mass of the parent droplet m_{new} (SI unit: kg) is

$$m_{new} = m_0 - n_{ch}r_{ch}$$

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. 30, the wavelength of the fastest-growing disturbance Λ_{RT} (SI unit: m) is

$$\Lambda_{RT} = 2\pi \sqrt{\frac{3\sigma}{a\rho_p}}$$

The frequency of the fastest-growing wave Ω_{RT} (SI unit: 1/s) is

$$\Omega_{RT} = \sqrt{\frac{2a}{3}\sqrt{\frac{a\rho_p}{3\sigma}}}$$

where a (SI unit: m/s^2) is the magnitude of the droplet acceleration. The time constant for breakup τ_{RT} (SI unit: s) is

$$\tau_{RT} = \sqrt{\frac{3}{2a}\sqrt{\frac{3\sigma}{\rho_p a}}}$$

The radius of the child droplets r_{ch} (SI unit: m) is

$$r_{ch} = \frac{C_{RT} \Lambda_{RT}}{2}$$

where C_{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_{RT} (SI unit: s) is defined as an auxiliary dependent variable for all particles. Whenever $t_{RT} > \tau_{RT}$ at the beginning of a time step taken by the solver, the parent droplet is broken up into child droplets. The number of child droplets n_{ch} is

$$n_{ch} = \text{floor}\left(\frac{r_0^3}{r_{ch}^3}\right)$$

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_{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_{KH} and the time since previous Rayleigh-Taylor breakup t_{RT}.

If the condition for Kelvin-Helmholtz breakup is met, then for the parent droplet and all child droplets t_{RT} is set to zero.

If the condition for Rayleigh-Taylor breakup is met, then for all child droplets the Kelvin-Helmholtz radius r_{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. 31). This model requires the following assumptions:

- The inlet is circular and produces a round liquid jet of density ρ_p (SI unit: 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_{inj} (SI unit: Pa) and the ambient pressure p_{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 Θ (SI unit: rad) is constant and stays constant as the injection progresses.

> ⑦ The following discussion uses several dimensionless variables. Their definitions are listed in Table 5-2 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 Θ. 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 Λ (SI unit: m) and Ω (SI unit: 1/s) are, respectively, the wavelength and growth rate of the fastest-growing disturbance on the surface of a liquid jet:

$$\Lambda_{KH} = \frac{9.02 r_0 (1 + 0.45\sqrt{Z_l})(1 + 0.4 T^{0.7})}{(1 + 0.865 We_g^{1.67})^{0.6}}$$

$$\Omega_{KH} = \frac{0.34 + 0.385 We_g^{1.5}}{(1 + Z_l)(1 + 1.4 T^{0.6})}\sqrt{\frac{\sigma}{\rho_p r_0^3}}$$

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 $t_1, t_2, t_3, ..., t_{N_t}$, 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 Δt_i are defined as

$$\Delta t_i = \begin{cases} \dfrac{t_2 - t_1}{2} & i = 1 \\[2mm] \dfrac{t_{i+1} - t_{i-1}}{2} & 1 < i < N \\[2mm] \dfrac{t_N - t_{N-1}}{2} & i = N \end{cases}$$

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

$$m_p C_p \frac{dT_p}{dt} = Q \tag{5-16}$$

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 5-16:

$$Q = hA_{\mathrm{p}}(T - T_{\mathrm{p}})$$

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

$$\mathrm{Bi} = \frac{hL_{\mathrm{C}}}{k_{\mathrm{p}}}$$

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 5-16:

$$Q = \varepsilon_{\mathrm{p}}\sigma A_{\mathrm{p}}(T^4 - T_{\mathrm{p}}^4)$$

where

- ε_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 5-16.

Computing Particle Mass or Diameter

You can release a distribution of particle sizes by selecting one of the options **Specify particle diameter** or **Specify particle mass** from the Particle Size Distribution list in the physics interface **Additional Variables** section. While solving for particle diameter or mass, a text field for the **Accretion rate** R (SI unit: kg/s) is shown in the settings window for the Particle Properties feature.

The accretion rate is the time derivative of particle mass. If the particle mass m_p (SI unit: kg) is solved for, its governing equation is

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

If the particle diameter d_p (SI unit: m) is solved for, its governing equation is

$$\frac{dd_p}{dt} = \frac{2R}{\pi \rho_p d_p^2}$$

where ρ_p (SI unit: kg/m^3) is the particle density.

Velocity of Mass Gained or Lost

When the particle diameter or mass is solved for by selecting **Specify particle diameter** or **Specify particle mass**, the interpretation of Newton's second law of motion for each particle is

$$m_p \frac{d}{dt}\left(\frac{d\mathbf{q}}{dt}\right) = \mathbf{F}_t \tag{5-17}$$

instead of the more customary, momentum-conserving expression

$$\frac{d}{dt}\left(m_p \frac{d\mathbf{q}}{dt}\right) = \mathbf{F}_t \qquad (5\text{-}18)$$

The physical interpretation of Equation 5-17 is that the mass gained or lost by the model particle is moving with the particle's velocity, whereas the physical interpretation of Equation 5-18 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 solving for particle diameter or mass, Equation 5-17 is solved. To instead solve Equation 5-18, so that mass gained or lost by the particle affects its velocity such that the momentum of the model particle is conserved, select **Uniform size** from the **Particle size distribution** list and instead either enter a time-dependent expression for the particle mass directly in the settings for the **Particle Properties** node, or express the particle mass in terms of a user-defined auxiliary dependent variable.

Types of Particle Size Distribution

When releasing particles, it is possible to specify a list of particle diameters or to sample the particle diameters from a distribution function. The built-in distribution functions include normal, lognormal, and uniform distributions.

NORMAL DISTRIBUTION
For a normal distribution, the probability distribution function is

$$f(d_p) = \frac{1}{\sigma\sqrt{2\pi}}\exp\left[-\frac{(d_p-\mu)^2}{2\sigma^2}\right]$$

where

- d_p (SI unit: m) is the particle diameter,
- μ (SI unit: m) is the average particle diameter, and
- σ (SI unit: m) is the standard deviation of the particle diameter.

In this context, the average is an arithmetic mean over released particles,

$$\mu = \frac{\sum d_p}{N} \qquad (5\text{-}19)$$

The standard deviation is

$$\sigma = \sqrt{\frac{\sum (d_p - \mu)^2}{N}} \qquad (5\text{-}20)$$

Because of the way initial values of auxiliary dependent variables are sampled in particle release features, Equation 5-19 will be correct to an extremely high degree of precision even if the number of particles is relatively small. In contrast, Equation 5-20 will only approximately be respected, with the standard deviation over released particles approaching the specified value σ as the number of particles N is increased.

LOGNORMAL DISTRIBUTION

A lognormal distribution of released particle diameters follows the probability distribution function (Ref. 32)

$$f(d_p) = \frac{1}{d_p \log \text{GSD} \sqrt{2\pi}} \exp\left[-\frac{(\log d_p - \log \text{CMD})^2}{2\log^2 \text{GSD}}\right]$$

where CMD (SI unit: m) is the count median diameter and GSD (dimensionless) is the geometric standard deviation. In general GSD should be greater than 1. By definition, half of the released particles have diameter greater than the count median diameter.
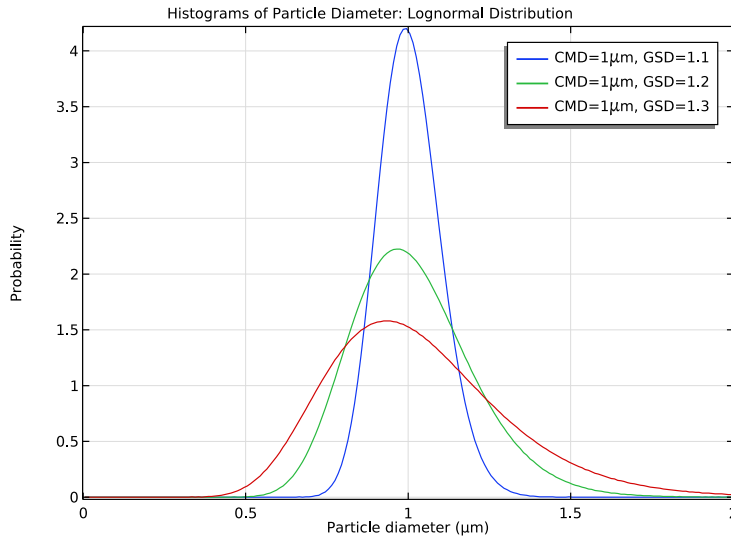
Figure 5-9: Probability distribution function of particle diameter for three different values of the geometric standard deviation.

The relationship between the geometric standard deviation GSD and the standard deviation from Equation 5-20, simply labeled σ, is

$$\sigma = \mu\sqrt{\exp(\log^2 \text{GSD}) - 1}$$

For a lognormal distribution, the mean diameter μ, count median diameter CMD, and Sauter mean diameter d_{sm} are related by

$$\mu = \text{CMD}\exp(0.5\log^2\text{GSD})$$
$$d_{sm} = \text{CMD}\exp(2.5\log^2\text{GSD})$$

The Sauter mean diameter is the diameter of a particle that has the same volume to surface area ratio as the population of particles as a whole,

$$d_{sm} = \frac{\sum d_p^3}{\sum d_p^2}$$

The equations relating CMD, μ, and d_{sm} are special cases of a general class of conversion equations known as the Hatch-Choate Equations, following a general form

$$d_A = \text{CMD}\exp(b\log^2\text{GSD})$$

where d_A is some type of average diameter yet to be specified and b is a dimensionless constant. The values of b for some typical average quantities are given in Table 5-3.

TABLE 5-3: DEFINITIONS OF VARIOUS AVERAGE PARTICLE DIAMETERS, AND THEIR COEFFICIENTS FOR THE HATCH-CHOATE EQUATIONS.

QUANTITY	EXPRESSION	COEFFICIENT
Count median diameter	50% of particles have diameter greater than the CMD.	0
Count mean diameter	$\bar{d} = \dfrac{\sum d_p}{N}$	0.5
Diameter of average surface area	$d_{\bar{s}} = \left(\dfrac{\sum d_p^2}{N}\right)^{1/2}$	1

TABLE 5-3: DEFINITIONS OF VARIOUS AVERAGE PARTICLE DIAMETERS, AND THEIR COEFFICIENTS FOR THE HATCH-CHOATE EQUATIONS.

QUANTITY	EXPRESSION	COEFFICIENT
Length median diameter	The sum of the diameters of particles larger than the LMD is 50% of the sum of all diameters.	1
Diameter of average mass	$d_{\bar{m}} = \left(\dfrac{\sum d_p^3}{N} \right)^{1/3}$	1.5
Length mean diameter	$d_{lm} = \dfrac{\sum d_p^2}{\sum d_p}$	1.5
Surface median diameter	The total surface area of particles larger than the SMD is 50% of the total surface area.	2
Sauter mean diameter (surface area mean diameter)	$d_{sm} = \dfrac{\sum d_p^3}{\sum d_p^2}$	2.5
Mass median diameter	The amount of mass in particles larger than the MMD is 50% of the total mass.	3
Mass mean diameter	$d_{mm} = \dfrac{\sum d_p^4}{\sum d_p^3}$	3.5

Charge Accumulation

The Charge Accumulation feature is used to compute the charge accumulated at a surface of a particle in a background of ions as a function of time. Charging models can be classified according to charging polarity (unipolar or bipolar), aerosol regime (continuum, free molecular, or transition regime), and charging mechanism (diffusion charging and/or field charging). The charging models used in this feature are unipolar, meaning that the ions in the background are either positively or negatively charged. The different models categories are summarized in Table 5-4 as a function of the particle diameter for STP conditions. The Knudsen number is defined as $Kn = \lambda / r_p$, where λ (SI unit: m) is the ion mean free path and r_p (SI unit: m) is the particle radius.

For reference, the mean free path at atmospheric pressure of small ions such as oxygen and nitrogen is about 100 nm, and of larger ions such as water and methane agglomerates is around 10 nm.

TABLE 5-4: CHARGING REGIMES AND MODELS USED AS A FUNCTION OF THE PARTICLE RADIUS AT STP

FREE MOLECULAR	TRANSITION	CONTINUUM
Kn>>1	Kn~1	Kn<<1
0.5 nm	5 nm - 50 nm	0.5 µm
Kinetic models	Limiting sphere models	Diffusion and field charging models

Ionic charging of particles is usually described in terms of diffusion charging or field charging. *Diffusion charging* occurs when the particle acquires a charge by virtue of the random thermal motion of ions and their collision and adherence to the particles. *Field charging* occurs when the electric flux lines deflect toward the particle resulting in the capture of ions. As the particle becomes charged, ions begin to be repelled by the particle, reducing the rate of charging. Eventually, the particle will reach a saturation charge and charging will cease (Ref. 33).

Diffusion charging is the dominant mechanism for small particles or low fields, while field charging is dominant for large particles and high fields. At intermediate sizes, diffusion and field charging can coexist; a simple way to estimate the total charging rate is to simply add the contributions from the two charging mechanisms. Lawless (Ref. 34) developed a more elaborate model to combine diffusion and field charging that reduces the diffusional component as the field becomes stronger, giving a distinctive diffusion charging rate above the saturation charge.

All the charging models here presented can increase the charge of a particle by fractional amounts of the elementary charge. This presents a problem for particles small enough that physically can only collect one charged ion or none. That is, the quantization of charge is neglected.

The charging models available in this feature are for the continuum and free molecular regime. Currently, this feature does not include models for the transition regime. The models are described in the below. To define the models the following dimensionless quantities are used:

• Particle charge

$$v_e = \frac{|Z|e^2}{4\pi\varepsilon_0 r_p k_B T_i}$$

- Characteristic charging time

$$\tau_c = \frac{e^2}{4\pi\rho_q \mu_i r_p k_B T_i}$$

- Electric field

$$w_e = \frac{r_p |\mathbf{E}|}{k_B T_i / e}.$$

The saturation charge of a dielectric particle is defined as

$$v_s = 3w_e \frac{\varepsilon_{r,p}}{\varepsilon_{r,p} + 2}$$

and the saturation charge of a conductive particle is defined as

$$v_s = 3w_e.$$

CLASSICAL DIFFUSION CHARGING MODEL

$$\tau_c \frac{dZ}{dt} = R_d$$

$$R_d = \frac{v_e}{\exp(v_e) - 1}$$

CLASSICAL FIELD CHARGING MODEL

$$\tau_c \frac{dZ}{dt} = R_f$$

$$R_f = \begin{cases} \dfrac{v_s}{4\varepsilon_0}\left(1 - \dfrac{v_e}{v_s}\right)^2 & |v_e| \le |v_s| \\ 0 & |v_e| > |v_s| \end{cases}$$

CLASSICAL DIFFUSION AND FIELD CHARGING MODEL

$$\tau_c \frac{dZ}{dt} = R_d + R_f$$

Where R_d and R_f use the same definitions as the **Classical diffusion** and **Classical field** charging models, respectively.

LAWLESS CHARGING MODEL

$$\tau_c \frac{dZ}{dt} = \begin{cases} R_{f,L} + f_a & |v_e| \leq |v_s| \\ R_{d,L} f_a & |v_e| > |v_s| \end{cases}$$

$$R_{f,L} = \frac{v_s}{4\varepsilon_0}\left(1 - \frac{v_e}{v_s}\right)^2 \qquad R_{d,L} = \frac{v_e - v_s}{\exp(v_e - v_s) - 1}$$

$$f_a = \begin{cases} \dfrac{1}{(w_e + 0,475)^{0,575}} & w_e \geq 0,525 \\ 1 & w_e < 0,525 \end{cases}$$

WHITE CHARGING MODEL

White's charging model is one of the simplest models for the free molecular regime. Using kinetic arguments and assuming a Boltzmann distribution to describe the ion density around a particle it is possible to deduce the rate of charge accumulation to be

$$\frac{dZ}{dt} = \frac{\pi \rho_q v_{th} r_p^2}{e} \exp\left(-\frac{|Z|e^2}{4\pi\varepsilon_0 r_p k_B T_i}\right)$$

where the ion thermal velocity is given by

$$v_{th} = \sqrt{\frac{8 k_B T_i}{m_i \pi}}.$$

The following definitions are used in this section:

- Z (dimensionless) is the total charge of the particle,
- $\varepsilon_{r,p}$ (dimensionless) is the particle permitivity,
- r_p (SI unit: m) is the particle radius,
- T_i (SI unit: K) is the ion temperature,
- m_i (SI unit: K) is the ion mass,
- $\mathbf{E}$ (SI unit: V/m) is the electric field,
- e (SI unit: C) is the elementary charge, and
- ρ_q (SI unit: C) is the ion charge density.

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. J. Happel and H. Brenner. *Low Reynolds number hydrodynamics: with special applications to particulate media*, vol. 1. Springer Science & Business Media, 2012.

6. S.G. Jennings, "The mean free path in air", *Journal of Aerosol Science*, vol. 19, no. 2 (1988), pp. 159–166.

7. M.D. Allen and O. G. Raabe, "Re-evaluation of Millikan's oil drop data for the motion of small particles in air", *Journal of Aerosol Science*, vol. 13, no. 6 (1982), pp. 537–547.

8. R.A. Millikan, "Coefficients of slip in gases and the law of reflection of molecules from the surfaces of solids and liquids." *Physical review*, vol. 21, no. 3 (1923), pp. 217–238.

9. R.A. Millikan, "The general law of fall of a small spherical body through a gas, and its bearing upon the nature of molecular reflection from surfaces", *Physical review*, vol. 22, no. 1 (1923), pp. 1–23.

10. P.S. Epstein, "On the resistance experienced by spheres in their motion through gases", Physical Review, vol. 23, no. 6 (1924), pp. 710–733.

11. W.F. Phillips, "Drag on a small sphere moving through a gas", *The Physics of Fluids*, vol. 18, No. 9 (1975), pp. 1089–1093.

12. L. Tian and G. Ahmadi, "Particle deposition in turbulent duct flows–comparisons of different model predictions", *Aerosol Science*, vol. 38, 2007, pp. 377–397.

13. P.G.T. Saffman, "The lift on a small sphere in a slow shear flow", *J. Fluid Mech.* 22, no. 02 (1965), pp. 385–400. Corrigendum, ibid. 31, no. 03 (1968), p. 624.

14. B.P. Ho and L.G. Leal, "Inertial migration of rigid spheres in two-dimensional unidirectional flows", *J. Fluid Mech.* 65, no. 02 (1974), pp. 365–400.

15. 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.

16. 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.

17. P.A. Durbin, "Stochastic Differential Equations and Turbulent Dispersion", NASA Reference Publication 1103, 1983.

18. A.D. Gosman and E. Ioannides, "Aspects of computer simulation of liquid-fueled combustors", *Journal of Energy*, vol. 7, no. 6, 1983, pp. 482–490.

19. M.R. Maxey and J.J. Riley, "Equation of motion for a small rigid sphere in a nonuniform flow", *The Physics of Fluids*, vol. 26, no. 4, 1983, pp. 883–889.

20. 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.

21. 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.

22. L.P. Gorkov, *Sov. Phys. Doklady*, vol. 6, p. 773, 1962.

23. J.T. Karlsen and H. Bruus, Forces acting on a small particle in an acoustical field in a viscous fluid, Phys. Rev. E, vol. 85, no. 1, 2012: 016327.

24. J.T. Karlsen and H. Bruus, Forces acting on a small particle in an acoustical field in a thermoviscous fluid, Phys. Rev. E, vol. 92, no. 4, 2015: 043010.

25. 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.

26. 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.

27. 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.

28. I. Finnie, "Some Observations on the Erosion of Ductile Metals", *Wear*, vol. 19, pp. 81–90, 1972.

29. 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.

30. M.A. Patterson and R.D. Reitz, *Modeling the Effects of Fuel Spray Characteristics on Diesel Engine Combustion and Emission*, SAE Paper 980131, 1998.

31. R.D. Reitz, *Modeling Atomization Processes in High-Pressure Vaporizing Sprays*, Atomization and Spray Technology, vol. 3, pp. 309–337, 1987.

32. W.C. Hinds, *Aerosol Technology*, Wiley, 1999.

33. R.C. Flanagan and J.H. Seinfeld, *Fundamental of Air Pollution Engineering*, Prentice Hall, 1998.

34. P.A. Lawless, *Particle Charging Bounds, Symmetry Relations, and an Analytic Charging Rate Model for the Continuum Regime*, J. Aerosol Sci., vol. 27, pp.191–215, 1996.

6

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 nonrelativistic 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 Couplings** 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, an empty **Multiphysics Couplings** node appears. 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 Couplings** 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 **Particle Release and Propagation**, the **Particle release specification** is set to **Specify current**.
	The **Electric Force** node is added to the Charged Particle Tracing interface. The **Domain Selection** is the same as that of the Charged Particle Tracing interface.
Electric Particle Field Interaction	The **Domain Selection** is the same as that of the participating physics interfaces.
	The corresponding Electrostatics and Charged Particle Tracing interfaces are preselected in the **Electric Particle Field Interaction** 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 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 1×10^6. 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** β (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 β. This improves the probability of stable convergence for models in which overestimation of the space charge density may lead to nonphysical 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 nonphysical 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 β, 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 Workflow 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.

	In order to use the **Space Charge Limited Emission** node, **Specify current** must be selected from the **Particle release specification** list in the coupled Charged Particle Tracing 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 scle1.

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 1000.

EXTRA DIMENSION SETTINGS

Enter the **Position offset** o_s (SI unit: m). The default value is 1 mm. This should be smaller than the length scale of the model geometry, but if it is several orders of magnitude smaller, it may become difficult for the model to converge.

Select the **Initial value of electric potential in extra dimension** — **Automatic** (the default) or **User defined**. For **User defined** enter the value for $V_{\mathrm{xd},0}$ (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 Workflow 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 nonrelativistic 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

$$-\nabla \cdot (\varepsilon_0 \nabla V - \mathbf{P}) = \rho$$

where

- $\varepsilon_0 = 8.854187817 \times 10^{-12}$ F/m is the permittivity of vacuum,
- V (SI unit: V) is the electric potential,
- $\mathbf{P}$ (SI unit: C/m$^{2)}$ is the polarization, and
- ρ (SI unit: C/m^3) 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

$$\frac{\mathrm{d}}{\mathrm{d}t}(m_\mathrm{p}\mathbf{v}) = -Ze\mathbf{E} + Ze(\mathbf{v} \times \mathbf{B})$$

where

- m_p (SI unit: kg) is the particle mass,
- $\mathbf{v}$ (SI unit: m/s) is the particle velocity,
- $e = 1.602176634 \times 10^{-19}$ C is the elementary charge,
- Z (dimensionless) is the particle charge number, and
- $\mathbf{B}$ (SI unit: T) is the magnetic flux density.

For nonrelativistic 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 ith source is denoted $\mathbf{q}_i$ (SI unit: m), the space charge density at position $\mathbf{r}$ is

$$\rho(\mathbf{r}) = \sum_{i=1}^{N} q_i \delta(\mathbf{r} - \mathbf{q}_i) \tag{6-1}$$

where δ is the Dirac delta function, q_i is the charge of the ith 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 alternative 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_{rel}.

The charged particles 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:

$$\frac{\mathrm{d}\rho(\mathbf{r})}{\mathrm{d}t} = \sum_{i=1}^{N} f_{\mathrm{rel},i} q_i \delta(\mathbf{r} - \mathbf{q}_i) \tag{6-2}$$

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, or until another user-specified convergence criterion has been satisfied.

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 ρ_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 ith 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_{rel}. The space charge within the mesh element can then be expressed as the solution to the first-order equation

$$\frac{d\rho_j}{dt} = \frac{1}{V_j} \sum_{i=1}^{N_j} f_{\mathrm{rel},i} q_i$$

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** β, 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

$$\bar{\rho}_s = \frac{\bar{\rho}_{s,\,prev} + \frac{iter}{\beta}\rho_s}{2}$$

where

- $\bar{\rho}_{s,\,prev}$ (SI unit: C/m^3) is the stored value of the cumulative space charge density from the previous iteration,

- ρ_s (SI unit: C/m^3) is the contribution of the particles to the space charge density in the current iteration,

- $iter$ (dimensionless) is the iteration number, and

- β (dimensionless) is the maximum number of ramping-up iterations.

CUMULATIVE AVERAGING STEP

After the ramping-up iterations are complete ($iter > \beta$), the cumulative space charge density at each subsequent iteration is

$$\bar{\rho}_s = \left(\sum_{j=1}^{i} w_j \right)^{-1} \left(\left(\sum_{j=1}^{i-1} w_j \right) \bar{\rho}_{s,\,prev} + w_i \rho_s \right)$$

$$i = \text{iter} - \beta + 1$$

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,

$$J = \frac{4\varepsilon_0}{9} \sqrt{\frac{2e}{m_e}} \frac{V^{3/2}}{d^2}$$

where

- $\varepsilon_0 = 8.854187817 \times 10^{-12}$ F/m is the permittivity of vacuum,
- $e = 1.602176634 \times 10^{-19}$ C is the elementary charge,
- $m_e = 9.10938356 \times 10^{-31}$ kg is the electron mass,
- 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 σ_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:

$$\frac{\partial^2 V_{xd}}{\partial x^2} = -\frac{\rho}{\varepsilon_0}$$

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

$$\frac{\partial^2 V_{xd}}{\partial x^2} = -\frac{J_0}{\varepsilon_0 \sqrt{2eV_{xd}/m_e}}$$

for which the following boundary conditions are enforced. At the cathode ($x = 0$),

$$V_{xd} = 0$$

$$-\mathbf{n} \cdot \varepsilon_0 \nabla V_{xd} = 0$$

At the emission surface ($x=\sigma_s$),

$$V_{xd} = V$$

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_{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_{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** σ_s is the width of the extra dimension in which V_{xd} is computed. As σ_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, σ_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** β 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 σ_s is reduced, it is often necessary to increase the **Number of iterations** β.

- 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 Couplings** 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 Couplings** 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 Couplings** 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 **Particle Release and Propagation**, the **Particle release specification** is set to **Specify current**, and the **Relativistic correction** check box is selected. The **Electric Force** and **Magnetic Force** nodes are added to the Charged Particle Tracing interface. The **Domain Selection** for each node is the same as that of the Charged Particle Tracing interface.

PHYSICS INTERFACE OR COUPLING FEATURE	MODIFIED SETTINGS (IF ANY)
Electric Particle Field Interaction	The **Domain Selection** is the same as that of the participating physics interfaces.
	The corresponding Electrostatics and Charged Particle Tracing interfaces are preselected in the **Electric Particle Field Interaction** section.
Magnetic Particle Field Interaction	The **Domain Selection** is the same as that of the participating physics interfaces.
	The corresponding Magnetic Fields and Charged Particle Tracing interfaces are preselected in the **Electric Particle Field Interaction** 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 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 mpfi1.

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 Workflow 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

$$-\nabla \cdot (\varepsilon_0 \nabla V - \mathbf{P}) = \rho$$

where

- $\varepsilon_0 = 8.854187817 \times 10^{-12}$ F/m is the permittivity of vacuum,
- V (SI unit: V) is the electric potential,
- $\mathbf{P}$ (SI unit: C/m^2) is the polarization, and
- ρ (SI unit: C/m^3) is the charge density.

The electric field **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 **A** (SI unit: Wb/m) may be written as

$$\sigma \frac{\partial \mathbf{A}}{\partial t} + \nabla \times \mathbf{H} = \mathbf{J}_{\text{e}}$$

$$\mathbf{B} = \nabla \times \mathbf{A}$$

where σ (SI unit: S/m) is the electrical conductivity, **H** (SI unit: A/m) is the magnetic field, **B** (SI unit: T) is the magnetic flux density, and $\mathbf{J}_{\text{e}}$ (SI unit: A/m^2) is the external current density. The constitutive relationship between **B** and **H** can be written

$$\mathbf{B} = \mu_0(\mathbf{H} + \mathbf{M})$$

where μ_0 (SI unit: H/m) is the permeability of free space and **M** (SI unit: A/m) is the magnetization.

The equation of motion for a charged particle in an electromagnetic field can be written as

$$\frac{\mathrm{d}}{\mathrm{d}t}(m_{\text{p}}\mathbf{v}) = -Ze\mathbf{E} + Ze(\mathbf{v} \times \mathbf{B})$$

where

- m_{p} (SI unit: kg) is the particle mass,
- **v** (SI unit: m/s) is the particle velocity,
- $e = 1.602176634 \times 10^{-19}$ C is the elementary charge,
- Z (dimensionless) is the particle charge number, and
- **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):

$$m_{\text{p}} = \frac{m_{\text{r}}}{\sqrt{1 - \mathbf{v} \cdot \mathbf{v}/c^2}}$$

where $c = 2.99792458 \times 10^8$ m/s is the speed of light in a vacuum. For nonrelativistic 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 ith 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

$$\mathbf{j}_s(\mathbf{r}) = \sum_{i=1}^{N} Z_i e \mathbf{v}_i \delta(\mathbf{r} - \mathbf{q}_i) \tag{6-3}$$

where

- δ is the Dirac delta function,
- Z_i (dimensionless) is the charge number of the ith particle,
- $e = 1.602176634 \times 10^{-19}$ C is the elementary charge,
- $\mathbf{v}_i$ (SI unit: m/s) is the velocity of the ith 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 alternative 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_{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:

$$\frac{d\mathbf{j}_s(\mathbf{r})}{dt} = \sum_{i=1}^{N} f_{rel,i} q_i \mathbf{v}_i \delta(\mathbf{r} - \mathbf{q}_i) \tag{6-4}$$

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

$$f_{rel} = \frac{I}{|q|N}$$

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, or until another user-specified convergence criterion has been satisfied.

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 ρ_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 ith 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_{rel}. Then the time derivative of the current density can be expressed as

$$\frac{d\mathbf{j}_{s,j}}{dt} = \frac{1}{V_j} \sum_{i=1}^{N_j} f_{rel,i} q_i \mathbf{v}_i$$

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 Couplings** 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 Multiphysics adds an empty **Multiphysics Couplings** 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 Couplings** 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 **Particle Release and Propagation**, the **Particle release specification** is set to **Specify mass flow rate**. The **Drag Force** node is added to the Particle Tracing for Fluid Flow interface. The **Domain Selection** is the same as that of the Particle Tracing for Fluid Flow interface.
Fluid-Particle Interaction	The **Domain Selection** 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 **Fluid-Particle Interaction** 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 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 Release and Propagation** 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 Workflow 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:

$$\rho \frac{\partial \mathbf{u}}{\partial t} + \rho(\mathbf{u} \cdot \nabla)\mathbf{u} = \nabla \cdot [-p\mathbf{I} + \mu(\nabla \mathbf{u} + (\nabla \mathbf{u})^{\mathrm{T}})] + \mathbf{F}$$

$$\rho \nabla \cdot \mathbf{u} = 0$$

where

- $\mathbf{u}$ (SI unit: m/s) is the fluid velocity,
- p (SI unit: Pa) is the pressure,
- ρ (SI unit: kg/m^3) is the density,

- μ (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

$$\frac{\mathrm{d}}{\mathrm{d}t}(m_\mathrm{p}\mathbf{v}) \; = \; \mathbf{F}_\mathrm{D} + \mathbf{F}_\mathrm{g} + \mathbf{F}_\mathrm{ext}$$

where

- m_p is the particle mass (SI unit: kg)
- $\mathbf{v}$ is the velocity of the particle (SI unit: m/s)
- $\mathbf{F}_\mathrm{D}$ is the drag force (SI unit: N),
- $\mathbf{F}_\mathrm{g}$ is the gravitational force (SI unit: N), and
- $\mathbf{F}_\mathrm{ext}$ 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 ith particle is denoted $\mathbf{q}_i$ (SI unit: m), the volume force at position $\mathbf{r}$ is

$$\mathbf{F}_V(\mathbf{r}) \; = \; -\sum_{i=1}^{N} \mathbf{F}_{\mathrm{D},i}\delta(\mathbf{r}-\mathbf{q}_i) \tag{6-5}$$

where δ is the Dirac delta function, $\mathbf{F}_{\mathrm{D},i}$ is the drag force exerted on the ith 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 alternative 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_{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:

$$\frac{d\mathbf{F}_V(\mathbf{r})}{dt} = -\sum_{i=1}^{N} f_{\text{rel}, i} \mathbf{F}_{D, i} \delta(\mathbf{r} - \mathbf{q}_i) \tag{6-6}$$

The volume force 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

$$f_{\mathrm{rel}} = \frac{\dot{m}}{m_{\mathrm{p}}N}$$

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, or until another user-specified convergence criterion has been satisfied.

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 ith 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_{rel}. The volume force within the mesh element can then be expressed as the solution to the first-order equation

$$\frac{d\mathbf{F}_{V,j}}{dt} = -\frac{1}{V_j} \sum_{i=1}^{N_j} f_{rel,i} \mathbf{F}_{D,i}$$

7

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

acoustophoretic radiation force See **acoustophoresis**.

acoustophoresis The migration of particles in spatially nonuniform acoustic pressure and acoustic velocity fields.

Brownian force See **brownian motion**.

Brownian motion The random drifting of particles suspended in a fluid due to the collisions of molecules with the particle surface.

Coulomb force The force between charged particles which is inversely proportional to the square of the distance between the particles.

dielectrophoretic force See **dielectrophoresis**.

dielectrophoresis The phenomenon in which a spatially nonuniform electric field exerts a force on a particle. Unlike the electric force, the dielectrophoretic force can be nonzero even if the particle is electrically neutral.

dispersed flow A fluid-particle system where particle-fluid and particle-particle interactions need to be accounted for in the model.

drag force The force exerted on a body by the surrounding fluid, in the direction perpendicular to the relative velocity of the fluid. Contrast with **lift force**.

electric force The force exerted on a charged particle by an electric field.

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.

hexapolar grid A grid of points consisting of uniformly spaced circular rings, each containing six more points the previous ring.

Kelvin-Helmholtz instability Unstable growth of waves on the surface of a fluid, such as a liquid droplet, or at the interface between two fluids arising from velocity shear. One of the mechanisms of droplet breakup.

Knudsen number The ratio of the mean free path of molecules in a gas to a representative length scale such as particle diameter; often used to classify the extent to which the surrounding gas can be modeled as a continuum flow.

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.

lift force The force exerted on a body by the surrounding fluid, in the direction perpendicular to the relative velocity of the fluid. Contrast with **drag force**.

magnetic force The force exerted on a moving charged particle by a magnetic field.

magnetophoretic force See **magnetophoresis**.

magnetophoresis The phenomenon in which a spatially nonuniform magnetic field exerts a force on a particle. Unlike the electric force, the dielectrophoretic force can be nonzero even if the particle is electrically neutral.

Maxwellian velocity distribution Describes the probability that a velocity is near a given value as a function of the temperature of the system.

Lorentz force The combined electric and magnetic force on a charged particle.

primary particle A model particle whose release is not contingent upon the existence of any other particle.

random number seed An argument to a pseudorandom number generator.

RANS Acronym for **Reynolds-averaged Navier-Stokes**.

Rayleigh-Taylor instability Instability at a fluid-fluid interface caused by normal acceleration of the less dense fluid in the direction of the denser fluid. One of the mechanisms of droplet breakup.

Reynolds-averaged Navier-Stokes Modification of the Navier-Stokes equations for fluid flow, in which a time-averaging operation has been performed on the equations of motion. The Reynolds' stresses (correlations between fluctuating velocity components) obtained from this averaging operation have to be obtained from an

additional set of equations, a closure. Turbulence models like the k-ε and Spalart-Allmaras models constitute closures to the RANS equations.

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.

secondary particle   A particle that is released due to an existing, currently active particle satisfying a given criterion, such as being subjected to a sufficiently high force or coming in contact with a surface.

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.

thermophoretic force   See **thermophoresis**.

thermophoresis   The phenomenon in which particles migrate in a spatially nonuniform temperature field.

transmission probability   The probability that a particle transmits from a given selected boundary or domain to another.

turbulent dispersion   The random diffusion of particles in a turbulent flow due to the random creation and annihilation of eddies in the flow.

wall condition   The effect of contact with a boundary on a particle's motion. Absorption and specular reflection are examples of wall conditions.

Index