Particle Tracing Module

User´s Guide

VERSION 4.3

COMSOL

Particle Tracing Module User's Guide

Version: May 2012 COMSOL 4.3

Contact Information

Visit www.comsol.com/contact for a searchable list of all COMSOL offices and local representatives. From this web page, search the contacts and find a local sales representative, go to other COMSOL websites, request information and pricing, submit technical support queries, subscribe to the monthly eNews email newsletter, and much more.

If you need to contact Technical Support, an online request form is located at www.comsol.com/support/contact.

Other useful links include:

- Technical Support www.comsol.com/support
- Software updates: www.comsol.com/support/updates
- Online community: www.comsol.com/community
- Events, conferences, and training: www.comsol.com/events
- Tutorials: www.comsol.com/products/tutorials
- Knowledge Base: www.comsol.com/support/knowledgebase

Part No. 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: Glossary

1

Introduction

This guide describes the Particle Tracing Module, an optional add-on package for COMSOL Multiphysics designed to assist you to solve and model field particle interactions. The module can also be used to compute particle trajectories where the particle motion does not affect the field.

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
- Particle Tracing Module Physics Guide
- Show More Physics Options
- Where Do I Access the Documentation and Model Library?
- Typographical Conventions

What Can the Particle Tracing Module Do?

The Particle Tracing Module is a general purpose, flexible tool that allows you to trace the trajectories of particles in the presence of an external field.

A custom interface is available for modeling the trajectory of electrons and ions in an electromagnetic field. 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. It is also possible to model particle-field interactions provided that the particle trajectories and electromagnetic fields are computed simultaneously.

There is also a dedicated interface for tracing particle trajectories in a fluid system. There are predefined forces for drag, gravity, and other forces, which can be conveniently be added to your model. Fluid-based particle tracing allows for detailed investigation of mixing or separation devices. Furthermore, you can study complicated physical processes such as electrophoresis, dielectrophoresis, magnetophoresis, and acoustophoresis by adding these predefined forces to your model. Finally, fluid-particle interaction is also possible to model, provided that the particle trajectories and fluid flow are computed simultaneously.

A mathematical-based particle tracing interface gives you complete freedom and flexibility over the equations of motion governing particle trajectories. There are a number of different formulations available that makes it possible to specify, for example, a Lagrangian or Hamiltonian that dictates the motion of the particles.

All of the interfaces provide many different ways of releasing particles, either on the boundary or domain level. Particles can also be released at multiple different times. There are several options available for describing how the particles interact with the wall, including specular reflection, sticking probabilities and secondary particle emission. The interfaces can all be used with COMSOL Multiphysics, but tracing charged particles benefits from the AC/DC Module because it makes it possible to model complex alternating or direct current systems. Similarly, tracing particles in a fluid system benefits from the CFD Module or the Microfluidics Module due to the number of advanced fluid flow features available in those modules.

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

The Particle Tracing Module has some 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.

See Also

- Particle Trajectories Study in the *COMSOL Multiphysics Reference Guide*

In the *COMSOL Multiphysics User's Guide:*

- Particle Data Sets
- Particle Trajectories Plots
- Phase Portrait Plots
- Poincaré Map Plots
- Particle Evaluation

Important

The links to the features described in the *COMSOL Multiphysics User's Guide* do not work in the PDF, only from the online help in COMSOL Multiphysics.

Tip

To locate and search all the documentation, in COMSOL, select **Help>Documentation** from the main menu and either enter a search term or look under a specific module in the documentation tree.

Model

For an example of Particle Trajectories Plots see Ion Cyclotron Motion: Model Library path: **Particle_Tracing_Module/Tutorial_Models/ion_cyclotron_motion**

For an example of Poincaré Map Plots see Rössler Attractor: Model Library path **Particle_Tracing_Module/Tutorial_Models/rossler_attractor**

For an example of Phase Portrait Plots see Ideal Cloak: Model Library path **Particle_Tracing_Module/Ray_Tracing/ideal_cloak**

Particle Tracing Module Physics Guide

See Also

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

PHYSICS	ICON	TAG	SPACE DIMENSION	PRESET STUDIES
AC/DC				
Charged Particle Tracing		cpt	3D, 2D, 2D axisymmetric	time dependent
Fluid Flow				
Particle Tracing for Fluid Flow		fpt	3D, 2D, 2D axisymmetric	time dependent
Mathematics				
Mathematical Particle Tracing		pt	3D, 2D, 2D axisymmetric	time dependent

Show More Physics Options

There are several features available on many physics interfaces or individual nodes. This section is a short overview of the options and includes links to the *COMSOL*

Multiphysics User's Guide or *COMSOL Multiphysics Reference Guide* where additional information is available.

Important

The links to the features described in the *COMSOL Multiphysics User's Guide* and *COMSOL Multiphysics Reference Guide* do not work in the PDF, only from within the online help.

Tip

To locate and search all the documentation for this information, in COMSOL, select **Help>Documentation** from the main menu and either enter a search term or look under a specific module in the documentation tree.

To display additional features for the physics interfaces and feature nodes, click the **Show** button ( ) on the **Model Builder** and then select the applicable option.

After clicking the **Show** button ( ), some sections display on the settings window when a node is clicked and other features are available from the context menu when a node is right-clicked. For each, the additional sections that can be displayed include **Equation**, **Advanced Settings**, **Discretization**, **Consistent Stabilization**, and **Inconsistent Stabilization**.

You can also click the **Expand Sections** button ( ) in the **Model Builder** to always show some sections or click the **Show** button ( ) and select **Reset to Default** to reset to display only the **Equation** and **Override and Contribution** sections.

For most physics nodes, both the **Equation** and **Override and Contribution** sections are always available. Click the **Show** button ( ) and then select **Equation View** to display the **Equation View** node under all physics nodes in the **Model Builder**.

Availability of each feature, and whether it is described for a particular physics node, is based on the individual physics selected. For example, the **Discretization**, **Advanced**

Settings, **Consistent Stabilization**, and **Inconsistent Stabilization** sections are often described individually throughout the documentation as there are unique settings.

SECTION	CROSS REFERENCE	LOCATION IN COMSOL MULTIPHYSICS USER GUIDE OR REFERENCE GUIDE
Show More Options and Expand Sections	• Showing and Expanding Advanced Physics Sections • The Model Builder Window	User's Guide
Discretization	• Show Discretization • Element Types and Discretization	User's Guide
	• Finite Elements • Discretization of the Equations	Reference Guide
Discretization - Splitting of complex variables	Compile Equations	Reference Guide
Pair Selection	• Identity and Contact Pairs • Specifying Boundary Conditions for Identity Pairs	User's Guide
Consistent and Inconsistent Stabilization	Show Stabilization	User's Guide
	• Stabilization Techniques • Numerical Stabilization	Reference Guide
Geometry	Working with Geometry	User's Guide
Constraint Settings	Using Weak Constraints	User's Guide

Where Do I Access the Documentation and Model Library?

A number of Internet resources provide more information about COMSOL Multiphysics, including licensing and technical information. The electronic documentation, Dynamic Help, and the Model Library are all accessed through the COMSOL Desktop.

! *Important*	If you are reading the documentation as a PDF file on your computer, the blue links do not work to open a model or content referenced in a different guide. However, if you are using the online help in COMSOL Multiphysics, these links work to other modules, model examples, and documentation sets.

THE DOCUMENTATION

The *COMSOL Multiphysics User's Guide* and *COMSOL Multiphysics Reference Guide* describe all interfaces and functionality included with the basic COMSOL Multiphysics license. These guides also have instructions about how to use COMSOL Multiphysics and how to access the documentation electronically through the COMSOL Multiphysics help desk.

To locate and search all the documentation, in COMSOL Multiphysics:

- Press F1 for Dynamic Help,
- Click the buttons on the toolbar, or
- Select **Help>Documentation** (?) or **Help>Dynamic Help** (?) from the main menu

and then either enter a search term or look under a specific module in the documentation tree.

THE MODEL LIBRARY

Each model comes with documentation that includes a theoretical background and step-by-step instructions to create the model. The models are available in COMSOL as MPH-files that you can open for further investigation. You can use the step-by-step instructions and the actual models as a template for your own modeling and applications.

SI units are used to describe the relevant properties, parameters, and dimensions in most examples, but other unit systems are available.

To open the Model Library, select **View>Model Library** ( ) from the main menu, and then search by model name or browse under a module folder name. Click to highlight any model of interest, and select **Open Model and PDF** to open both the model and the documentation explaining how to build the model. Alternatively, click the **Dynamic**

Help button ( ) or select **Help>Documentation** in COMSOL to search by name or browse by module.

The model libraries are updated on a regular basis by COMSOL in order to add new models and to improve existing models. Choose **View>Model Library Update** ( ) to update your model library to include the latest versions of the model examples.

If you have any feedback or suggestions for additional models for the library (including those developed by you), feel free to contact us at info@comsol.com.

CONTACTING COMSOL BY EMAIL

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

To receive technical support from COMSOL for the COMSOL products, please contact your local COMSOL representative or send your questions to support@comsol.com. An automatic notification and case number is sent to you by email.

COMSOL WEB SITES

Main Corporate web site	www.comsol.com
Worldwide contact information	www.comsol.com/contact
Technical Support main page	www.comsol.com/support
Support Knowledge Base	www.comsol.com/support/knowledgebase
Product updates	www.comsol.com/support/updates
COMSOL User Community	www.comsol.com/community

Typographical Conventions

All COMSOL user's guides use a set of consistent typographical conventions that make it easier to follow the discussion, understand what you can expect to see on the graphical user interface (GUI), and know which data must be entered into various data-entry fields.

In particular, these conventions are used throughout the documentation:

CONVENTION	EXAMPLE
text highlighted in blue	Click text highlighted in blue to go to other information in the PDF. When you are using the online help desk in COMSOL Multiphysics, these links also work to other modules, model examples, and documentation sets.
boldface font	A **boldface** font indicates that the given word(s) appear exactly that way on the COMSOL Desktop (or, for toolbar buttons, in the corresponding tip). For example, the **Model Builder** window ( ) is often referred to and this is the window that contains the model tree. As another example, the instructions might say to click the **Zoom Extents** button ( ), and this means that when you hover over the button with your mouse, the same label displays on the COMSOL Desktop.
Forward arrow symbol **>**	The forward arrow symbol **>** is instructing you to select a series of menu items in a specific order. For example, **Options>Preferences** is equivalent to: From the **Options** menu, choose **Preferences**.
`Code` (monospace) font	A `Code` (monospace) font indicates you are to make a keyboard entry in the user interface. You might see an instruction such as "Enter (or type) `1.25` in the **Current density** field." The monospace font also is an indication of programming code or a variable name.
Italic *`Code`* (monospace) font	An italic *`Code`* (monospace) font indicates user inputs and parts of names that can vary or be defined by the user.
Arrow brackets <> following the `Code` (monospace) or *`Code`* (italic) fonts	The arrow brackets included in round brackets after either a monospace `Code` or an italic *`Code`* font means that the content in the string can be freely chosen or entered by the user, such as feature tags. For example, `model.geom(<`*`tag`*`>)` where <*`tag`*> is the geometry's tag (an identifier of your choice). When the string is predefined by COMSOL, no bracket is used and this indicates that this is a finite set, such as a feature name.

KEY TO THE GRAPHICS

Throughout the documentation, additional icons are used to help navigate the information. These categories are used to draw your attention to the information based on the level of importance, although it is always recommended that you read these text boxes.

ICON	NAME	DESCRIPTION
⚠	Caution	A Caution icon is used to indicate that the user should proceed carefully and consider the next steps. It might mean that an action is required, or if the instructions are not followed, that there will be problems with the model solution.
❗	Important	An Important icon is used to indicate that the information provided is key to the model building, design, or solution. The information is of higher importance than a note or tip, and the user should endeavor to follow the instructions.
📝	Note	A Note icon is used to indicate that the information may be of use to the user. It is recommended that the user read the text.
💡	Tip	A Tip icon is used to provide information, reminders, short cuts, suggestions of how to improve model design, and other information that may or may not be useful to the user.
🔍	See Also	The See Also icon indicates that other useful information is located in the named section. If you are working on line, click the hyperlink to go to the information directly. When the link is outside of the current PDF document, the text indicates this, for example See The Laminar Flow Interface in the *COMSOL Multiphysics User's Guide*. Note that if you are in COMSOL Multiphysics' online help, the link will work.
▥	Model	The Model icon is used in the documentation as well as in COMSOL Multiphysics from the View>Model Library menu. If you are working online, click the link to go to the PDF version of the step-by-step instructions. In some cases, a model is only available if you have a license for a specific module. These examples occur in the COMSOL Multiphysics User's Guide. The Model Library path describes how to find the actual model in COMSOL Multiphysics, for example If you have the RF Module, see Radar Cross Section: Model Library path **RF_Module/Tutorial_Models/radar_cross_section**
Space Dimension		Another set of icons are also used in the Model Builder—the model space dimension is indicated by 0D ·, 1D —, 1D axial symmetry ┤, 2D ▯, 2D axial symmetry ▯, and 3D ▣ icons. These icons are also used in the documentation to clearly list the differences to an interface, feature node, or theory section, which are based on space dimension.

Overview of the User's Guide

The *Particle Tracing Module User's Guide* gets you started with modeling using COMSOL Multiphysics. The information in this guide is specific to this module. Instructions how to use COMSOL in general are included with the *COMSOL Multiphysics User's Guide*.

Tip

As detailed in the section Where Do I Access the Documentation and Model Library? this information is also searchable from the COMSOL Multiphysics software **Help** menu.

TABLE OF CONTENTS, GLOSSARY, AND INDEX

To help you navigate through this guide, see the Contents, Glossary 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.

THE MATHEMATICS BRANCH

The Mathematical Particle Tracing Interface and its underlying theory is described in this chapter.

THE AC/DC BRANCH

The Charged Particle Tracing Interface and its underlying theory is described in this chapter.

THE FLUID FLOW BRANCH

The Particle Tracing for Fluid Flow Interface and its underlying theory is 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 interface for a specific problem.

In this chapter:

- Particle Tracing
- Modeling Tools

Particle Tracing

Particle tracing provides a Lagrangian description of a problem by solving ordinary differential equations using Newton's law of motion. Newton's law of motion requires specification of the particle mass, and all forces acting on the particle. The forces acting on particles 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.

For each particle, an 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. At each time step, 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. Since 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.

In this section:

- Charged Particle Tracing
- Particle Tracing for Fluid Flow
- Mathematical Particle Tracing

Charged Particle Tracing

The Charged Particle Tracing Interface is designed for modeling the motion of electrons, individual ions or small ion clusters in electric and magnetic fields.

SPACE CHARGE EFFECTS

If the number density of charged species is less than around 10^{13} $1/\mathrm{m}^3$, the effect of the particles on the fields can be neglected. This allows the fields to be computed in a different study from the particle trajectories. The fields may be computed using a Stationary, Frequency Domain or Time-dependent study. These fields are then used to determine the forces on particles, which are computed in a separate time-dependent study. The fact that the particle trajectories can be computed in their own study allows efficient and computationally inexpensive iterative solver to be used.

As the number density of the charged particles gets above $10^{13}\ 1/\mathrm{m}^3$, the particles will begin to exert a space charge on the field. This effect can be included in the model by adding a **Particle-Field Interaction** feature. The computational requirements for models which include particle-field interactions increase significantly over those which neglect it. The fields and particle trajectories must be computed in the same Time-dependent study and a direct solver must also be used, due to the fact that particle-field interactions produce an ill conditioned Jacobian matrix. In addition, the space charge produced by the charged particles is smeared out within the mesh element that the particle occupies, so results tend to be mesh dependent. For these reasons, consider neglecting space charge effects or including the Coulomb force directly as described below.

COULOMB FORCES

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

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 bubbles, particles, droplets etc. 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.

SPARSE FLOW

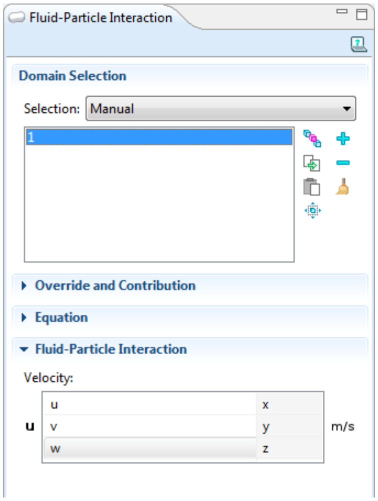
In a sparse flow, the continuous phase affects the motion of the particles but not vice-versa. This is often referred to as "one-way coupling.". When modeling such a system in COMSOL, it is usually most efficient to solve for the continuous phase and the dispersed phases in separate studies. For example, 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.

Model

Particle Trajectories in a Laminar Static Mixer: Model Library path **Particle_Tracing_Module/Fluid_Flow/laminar_mixer_particle**

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 "two-way coupling". In order to include this effect in COMSOL, a **Fluid-Particle Interaction** feature must be added to the model. The degrees of freedom from the continuous phase should be added in the **Velocity** text field.



Fluid-Particle Interaction

Domain Selection

Selection: Manual

1

Override and Contribution

Equation

Fluid-Particle Interaction

Velocity:

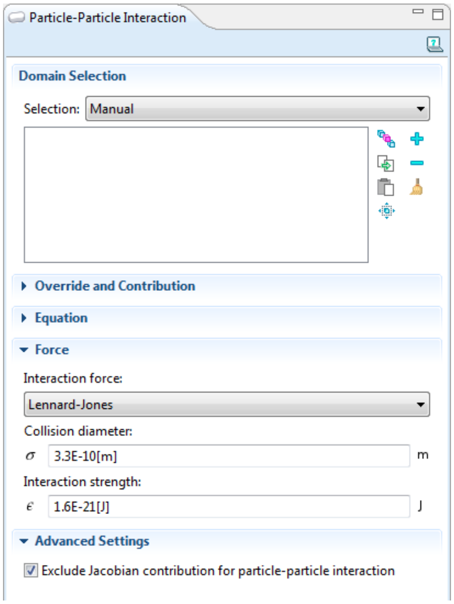
u	u	x	m/s
	v	y	
	w	z	

Figure 2-1: The Fluid-Particle Interaction settings window. In this case the degrees of freedom for the continuous phase are u, v, and w.

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 particle-fluid interaction feature somewhat mesh dependent. When modeling fluid-particle interaction, 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.

DISPERSED FLOW

In addition to the effects mentioned above, particle-particle interactions may 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** feature.



Selection: Manual

Interaction force: Lennard-Jones

Collision diameter: σ 3.3E-10[m] m

Interaction strength: ϵ 1.6E-21[J] J

☑ Exclude Jacobian contribution for particle-particle interaction

Figure 2-2: The Particle-Particle Interaction settings window.

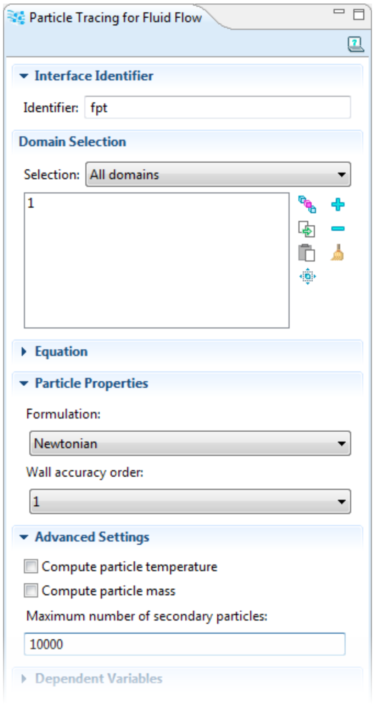
The following limitations apply:

- Hard sphere collisions are not supported. Forces must vary continuously with respect to the distance between particles as in, for example, the Coulomb force between charged particles.
- The computation time scales as the number of particles squared. This is because every particle interacts with every other particle over all distances. There is no way

to implement a cut-off scheme, where particles only see other particles within a certain radius.

- If the particle-particle interaction law is highly nonlinear, it may 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 will produce 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** option is selected, which preserves the sparseness in the Jacobian. Clearing this option is likely to result in a dramatic increase in the amount of memory and time to solve the problem.

The **Particle Tracing for Fluid Flow** interface provides a convenient way of computing the particle mass and temperature. This is activated in the properties settings window. When the option to compute particle mass is activated, it is possible to set the initial particle mass in the release and inlet features. 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.



Interface Identifier

Identifier: fpt

Domain Selection

Selection: All domains

1

Equation

Particle Properties

Formulation:

Newtonian

Wall accuracy order:

1

Advanced Settings

□ Compute particle temperature

□ Compute particle mass

Maximum number of secondary particles:

10000

Dependent Variables

Figure 2-3: Particle Tracing for Fluid Flow settings window.

MODELING ADVECTION AND DIFFUSION

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

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

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 Peclet number is above around 1000. The diffusion coefficient for spherical particles is:

$$D = \frac{k_B T}{6\pi\eta r}$$

where η is the fluid viscosity, k_B is Boltzmann's constant, T is the temperature, and r is the radius of the particles. For 100 nm diameter particles in water at room temperature, this results in a diffusion coefficient of around $4{\cdot}10^{-12}$ m^2/s. For a microfluidic device with characteristic size 1 mm and velocity of 1 m/s, this results in a Peclet number of $2.5{\cdot}10^8$. Handling such a large Peclet number with continuum methods is clearly not possible (Ref. 2).

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

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 doubles when the Hamiltonian formulation is activated.

Modeling Tools

In this section:

- Special Variables
- Monte Carlo Modeling
- Auxiliary Dependent Variables and Residence Time
- Filters

Special Variables

The particle tracing interfaces define a number of special variables, some of which can only be used during results processing. These variables can be found in the **Particle statistics** plot group during results processing, as shown in Figure 2-4.

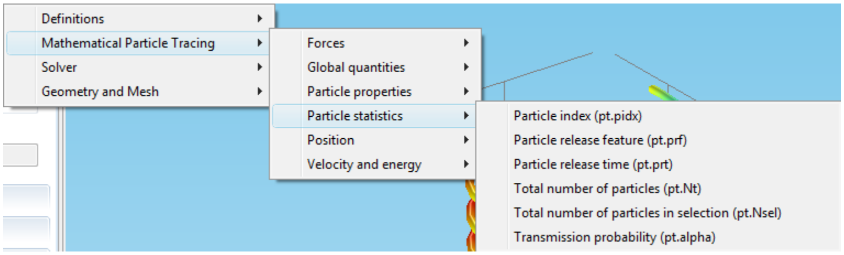

Definitions
Mathematical Particle Tracing
Solver
Geometry and Mesh
Forces
Global quantities
Particle properties
Particle statistics
Position
Velocity and energy
Particle index (pt.pidx)
Particle release feature (pt.prf)
Particle release time (pt.prt)
Total number of particles (pt.Nt)
Total number of particles in selection (pt.Nsel)
Transmission probability (pt.alpha)

Figure 2-4: An example of variables available from the particle statistics menu and available with the Mathematical Particle Tracing interface.

The following variables are defined:

- **Particle index**, `pt.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 which 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(pt.pidx,t)`.
- **Particle release feature**, `pt.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.

Model

When you also have the CFD Module, see Particle Tracing in a Micromixer: Model Library path **CFD_Module/Particle_Tracing/ micromixer_particle_tracing**

- **Particle release time**, `pt.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.

The following variables are only defined during results processing and can only be evaluated using the **Global Evaluation** node under **Derived values**. They cannot be plotted as the **Color Expression** in a **Particle Trajectories** plot, or used in the equation system.

- **Total number of particles**, `pt.Nt`. The total number of particles released may not be known in advance, especially in models which include secondary particle emission. The **Total number of particles** variable keeps track of this number.
- **Total number of particles in selection**, `pt.Nsel`. If a selection has been applied to the **Particle** data set, the number of particles in that selection can be evaluated.
- **Transmission probability**, `pt.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.

Model

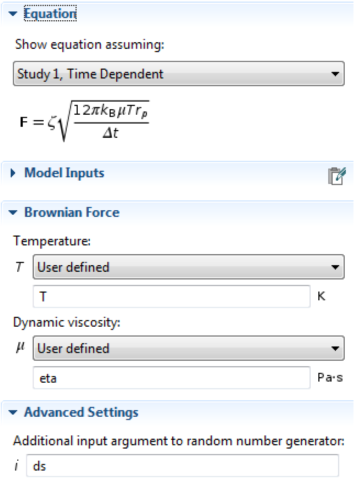
- Particle Trajectories in a Laminar Static Mixer: Model Library path **Particle_Tracing_Module/Fluid_Flow/laminar_mixer_particle**
- Brownian Motion: Model Libraty path **Particle_Tracing_Module/ Tutorial_Models/brownian_motion**

Monte Carlo Modeling

It is possible to add forces on particles which are stochastic. In the **Particle Tracing for Fluid Flow** interface, the **Drag Force** and **Brownian Force** features are forces that potentially include components which are random in nature. When these features are included in a model, it may be necessary to solve the problem multiple times and take some kind of statistical average of the results. Each time the model is solved, a new set of random numbers, which are used to compute the force, should be generated. The **Brownian Force** feature has a parameter called **Additional input argument to random number generator**. This value can be set by a parameter, which can then be set in a parametric sweep.

Model

Brownian Motion: Model Libraty path **Particle_Tracing_Module/ Tutorial_Models/brownian_motion**



Equation

Show equation assuming:

Study 1, Time Dependent

$$\mathbf{F} = \zeta\sqrt{\frac{12\pi k_B \mu T r_p}{\Delta t}}$$

Model Inputs

Brownian Force

Temperature:

T User defined

T K

Dynamic viscosity:

μ User defined

eta Pa·s

Advanced Settings

Additional input argument to random number generator:

i ds

Figure 2-5: An example of the Brownian Force settings window.

Auxiliary Dependent Variables and Residence Time

Auxiliary dependent variables can be used to help keep track of things like residence time, particle trajectory length, integrated shear rate etc. When an Auxiliary Dependent Variable feature is added to the physics interface an additional ordinary differential equation is solved for each particle. So, to compute the residence time of particles, do the following:

1 Right click on the particle tracing physics interface and select **Auxiliary Dependent Variable**.

2 Go to the **Settings** window for Auxiliary Dependent Variable.

3 Locate the **Auxiliary Dependent Variable** section. In the R edit field, type 1.

This will then solve the following differential equation for each particle:

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

and so the variable rp represents the residence time. In order to compute the length of the particle trajectories:

4 Locate the **Auxiliary Dependent Variable** section. From the **Integrate** list, choose **Along particle trajectory**.

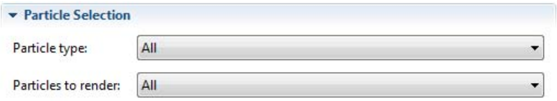
Now, the following differential equation is solved for each particle:

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

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.

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 on the **Particle Trajectories** plot type and choose **Filter**.



Particle type: All

Particles to render: All

Figure 2-6: Settings available in the Filter node.

The **Particle type** can be set to render primary particles, secondary particles or both. This option is only relevant when secondary particle emission is included in the model.

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.

3

Mathematical Particle Tracing

This chapter describes the Mathematical Particle Tracing interface found under the **Mathematics** branch (Δu) in the **Model Wizard**.

In this chapter:

The Mathematical Particle Tracing Interface

The **Mathematical Particle Tracing** interface ( ), found under the **Mathematics** branch (Δu) in the **Model Wizard**, computes the trajectories of particles through a geometry. The motion of the particles is usually driven by an external field. The particles can be massless, or have their motion determined by Newton's second law. Additionally, you can specify a Hamiltonian or Lagrangian that dictates the motion of the particles.

Model	• Rössler Attractor: Model Library path **Particle_Tracing_Module/Tutorial_Models/rossler_attractor** • Ion Cyclotron Motion: Model Library path: **Particle_Tracing_Module/Tutorial_Models/ion_cyclotron_motion**

When this interface is added, these default nodes are also added to the **Model Builder**—**Wall** and **Particle Properties**. Right-click the **Mathematical Particle Tracing** node to add other features.

INTERFACE IDENTIFIER

The interface identifier is a text string that can be used to reference the respective physics interface if appropriate. Such situations could occur when coupling this interface to another physics interface, or when trying to identify and use variables defined by this physics interface, which is used to reach the fields and variables in expressions, for example. It can be changed to any unique string in the **Identifier** field.

The default identifier (for the first interface in the model) is `pt`.

DOMAIN SELECTION

The default setting is to include **All domains** in the model to define the dependent variables and the equations. To choose specific domains, select **Manual** from the **Selection** list.

PARTICLE PROPERTIES

Select a **Formulation**—**Newtonian**, **Lagrangian**, **Massless**, or **Hamiltonian**. This selection changes the available sections displayed on the Particle Properties, Release from Grid, and Release settings windows. If **Newtonian** is selected, a Force node can also be added to the model.

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 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.

If the **Secondary emission** check box is selected in the **Wall** boundary condition, there is the potential to generate a large number of secondary particles. The **Maximum number of secondary particles** prevents an inordinate number of particles from being generated by capping them at the number supplied in the edit field.

The **Relativistic correction** option, applicable for particles with very high speed, means you can take relativistic effects on the particle mass into account. Select **On** from the **Relativistic correction** list (the default is to not use a relativistic correction). The particle mass m_p is then computed as

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

where $\mathbf{v}$ is the particle velocity and c is the speed of light.

MAXIMUM NUMBER OF SECONDARY PARTICLES

If the **Secondary emission** option is checked in the **Wall** boundary condition, there is the potential to generate a large number of secondary particles. The **Maximum number of secondary particles** prevents an inordinate number of particles from being generated by capping them at the number supplied in the edit field.

DEPENDENT VARIABLES

The dependent variables (field variables) are the **Particle position**, **Particle position components**, **Particle momentum**, and **Particle momentum components**. The name can be changed but the names of fields and dependent variables must be unique within a model.

See Also

- Show More Physics Options
- Theory for the Mathematical Particle Tracing Interface
- Domain, Boundary, Pair, and Global Conditions for the Mathematical Particle Tracing Interface

Domain, Boundary, Pair, and Global Conditions for the Mathematical Particle Tracing Interface

The Mathematical Particle Tracing Interface has these domain, boundary, pair, and global features available.

Domain

- Force—only available when **Newtonian** is selected as the **Formulation**
- Particle-Particle Interaction—only available when **Newtonian** is selected as the **Formulation**
- Release

Boundary

- Axial Symmetry
- Inlet
- Outlet
- Wall (default)

Pairs

- Inlet
- Particle Continuity

Global

- Auxiliary Dependent Variable
- Particle Properties (default, singleton)

- Release from Grid

See Also

The Weak Contribution Node in the *COMSOL Multiphysics User's Guide*

Force

Note

The **Force** node is only available when **Newtonian** is selected as the **Formulation** on the interface settings window.

Use the **Force** feature to define the force directly or using a susceptibility and field. It adds forces to the particles to influence the particle motion.

DOMAIN SELECTION

From the **Selection** list, choose the domains to apply the force.

FORCE

Select an option from the **Specify force** list—**Directly** or **Using susceptibility and field**.

- If **Directly** is selected, enter expressions for the **Force** $\mathbf{F}$ (SI unit: N) based on space dimension.
- If **Using susceptibility and field** is selected, select **Isotropic**, **Diagonal**, **Symmetric**, or **Anisotropic** and enter a value or expression for the **Susceptibility** χ and **Field** Γ in the field or matrix (based on space dimension). There are no units assigned to the **Susceptibility** and **Field** because there are an unlimited number of combinations whose product would result in the unit of force.

Particle-Particle Interaction

Note

The **Particle-Particle Interaction** node is only available when **Newtonian** is selected as the **Formulation** on the interface settings window.

Note

The **Particle-Particle Interaction** feature does not support hard sphere collisions.

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

DOMAIN SELECTION

From the **Selection** list, choose the domains to apply the force.

FORCE

Select an option from the **Interaction force** list—**Coulomb**, **Lennard-Jones**, or **User defined**.

- If **Coulomb** is selected, a Coulomb force describes the interaction between charged particles. No user input is necessary. This is the default option.
- If **Lennard-Jones** is selected, enter the **Collision diameter** (SI unit: m) and the **Interaction strength** (SI unit: J). The collision diameter is typically on the order of a few angstroms (the default value is $3.3 \cdot 10^{-10}$ m or 3.3 Å) and the interaction strength is usually on the order of 10^{-21} joules (the default value is $1.6 \cdot 10^{-21}$J). The value of these parameters depend on the gas molecules interacting and can usually be found from a literature search. This option uses the Lennard-Jones potential to approximate the interaction between neutral particles.
- If **User defined** is selected, enter a user-defined expression for the interaction force based on space dimension. There is specific syntax that you must use to specify user-defined interaction forces. 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 `<identifier>.r`, where `<identifier>` is the physics interface identifier. So, if the interface identifier is `pt`, then this variable is accessed using `pt.r`. As an example, the gravitational force on particle i depends on the position vector and mass of all other particles:

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

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

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

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

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. When the **Exclude Jacobian contribution for particle-particle interaction force** check box is selected, the contribution to the Jacobian matrix due to the particle-particle interaction force is ignored. This 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 may therefore need to take very small time steps when solving.

Model

Rotating Galaxy: Model Library path **Particle_Tracing_Module/Tutorial_Models/rotating_galaxy**

Release

Use the **Release** node to set the release times, initial position, and initial value of any auxiliary dependent variable. The sections available are based on the **Formulation** selected in the interface settings window. If **Newtonian**, **Lagrangian**, or **Hamiltonian** are selected, also enter an initial velocity.

RELEASE TIMES

Enter **Release times** 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 below.

INITIAL POSITION

Select an **Initial position**—**Mesh based** or **Density**.

- If **Mesh based** is selected the particles are released from a set of positions determined by a selection of geometric entities (of arbitrary dimension) in the mesh. Given a **Refinement factor** between 1 and 5, 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.
- If **Density** is selected, enter a value for the **Number of particles per release** N and the **Density proportional to** ρ. Both are unitless numbers. The **Number of particles per release** is the number before Maxwellian velocity multiplication. The field **Density proportional to** is an expression—the resulting particle distribution approximately has a density that is proportional to this expression. Evaluation of the density expression on the selected domains is taken as one value for each mesh element. The resulting distribution will look a bit random, and it will depend on the order in which the mesh elements are numbered. The distribution will probably not be exactly the same in different COMSOL versions, but the total number of particles released is always N. The **Density proportional to** expression must be strictly positive.

INITIAL VELOCITY

Note

For The Charged Particle Tracing Interface and The Particle Tracing for Fluid Flow Interfaces this section displays when **Newtonian** is selected as the **Formulation**.

For The Mathematical Particle Tracing Interface this section displays when **Newtonian**, **Lagrangian**, or **Hamiltonian** is selected as the **Formulation**.

Select an option from the **Initial velocity** list—**Expression** or **Maxwellian**.

- If **Expression** is selected, enter coordinates for the **Initial particle velocity** $\mathbf{v_0}$ (SI unit: m/s) based on space dimension.
- If **Maxwellian** is selected, enter the **Number of particles per velocity space direction** $N_\mathbf{v}$. The default is 50. Then enter the **Temperature** T_0 (SI unit: K). The default is 293.15 K. Note that this option can generate an overwhelming number of particles because for each release time and release position, this number is multiplied by N_v^{sdim}, where `sdim` is the number of space dimensions. Typically

this option should only be used when releasing particles at a single release time at a single point in space.

- If **Constant speed, spherical** is selected, enter the **Speed** v (SI unit: m/s) and the **Number of particles in velocity space** N_{vel}. This option generates N_{vel} particles for each release point, each with the same speed. The velocity distribution is spherical for 3D models and circular for 2D.
- If **Constant speed, hemispherical** is selected, enter the **Speed** v (SI unit: m/s), the **Number of particles in velocity space** N_{vel} and a direction vector for the **Hemisphere axis**. This option generates N_{vel} particles for each release point, each with the same speed. The velocity distribution is hemispherical for 3D models and a half circle for 2D. The hemisphere axis sets the direction of the north pole of the hemisphere or half circle.

INITIAL VALUE OF AUXILIARY DEPENDENT VARIABLES

If an Auxiliary Dependent Variable has been added to the model then there is an option to enter an **Initial value of auxiliary dependent variables**. There is an option to choose a distribution function for the initial value of the auxiliary dependent variables. In this case, the number of particles simulated may increase substantially and the following options are available. The **Distribution function** option is used to select whether the initial value of the auxiliary dependent variables should be a scalar value, or a distribution function.

- When the **Distribution function** is set to **None**, enter an initial value for each of the Auxiliary Dependent Variables added to the model.
- The **Normal** option creates a normal distribution function for the initial value of the auxiliary dependent variables, with a user defined **Mean** (default 0) and **Variance** (default 1). The **Number of values** sets the number of points in the distribution function.
- The **Lognormal** option creates a log-normal distribution function for the initial value of the auxiliary dependent variables, with a user defined **Mean** (default 0) and **Variance** (default 1). The **Number of values** sets the number of points in the distribution function.
- The **Uniform** option creates a uniform distribution function for the initial value of the auxiliary dependent variables, with a user defined **Mean** (default 0) and **Variance** (default 1). The **Number of values** sets the number of points in the distribution function.

Wall

Use the **Wall** feature to determine what happens to the particles when contact with a wall is made.

BOUNDARY SELECTION

From the **Selection** list, choose the **Boundaries** to set the wall.

WALL

Select a **Wall condition**—**Bounce**, **Freeze**, **Stick**, or **Disappear**.

- The **Bounce** option specularly reflects from the wall such that the particle momentum is conserved. This option is typically used when tracing microscopic particles in a fluid.
- The **Freeze** option (default) fixes 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.
- The **Stick** option fixes 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.
- The **Disappear** option means that the particle is not displayed once it has made contact with the wall. This option should be used if display of the particle location after contact with the wall is not of interest.

Select a **Primary particle condition**—**None**, **Probability** or **Expression**.

- The **None** option means that the option set for the **Wall condition** is always respected by the incident particles.
- The **Probability** option applies the **Wall condition** with a certain probability, γ, otherwise the particle bounces. 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, one will freeze and the remaining 9 particles will **Bounce**. If the **Wall condition** is set to **Stick** and γ is set to 0.5 then half of the particles will stick to the wall and the other half will bounce.
- The **Evaluation expression** option evaluates the expression, e, whenever the particle strikes the wall. If the **Evaluation expression** is zero, the particle behaves according to the **Wall condition**, otherwise the particle bounces.

Activate **Include secondary emission** to allow for secondary particles to be released when a particle strikes the wall. The following options become available.

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

Select an **Initial velocity—Isotropic hemisphere**, **Reflection of primary particle** or **User defined**.

- The **Isotropic hemisphere** option will release 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.
- The **Reflection of primary particle** option will release the secondary particles according to the law of specular reflection of the incident particle. Each secondary particle has the same speed equal to the speed of the primary particle divided by the number of secondary particles.
- The **User Defined** option allows for an arbitrary velocity vector to be set for the secondary particles. The **Specify tangential and normal velocity components** check box allows specification of one 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.

See Also

About the Wall Boundary Conditions

Axial Symmetry

2D Axi

An **Axial Symmetry** node is automatically added to 2D axisymmetric models. The options available for the feature are exactly the same as for the Wall, except the selection is locked and is only applicable on the symmetry axis. In addition, it is not possible to set the **Primary particle condition** or **Include secondary emission**.

Inlet

Use the **Inlet** feature to determine how particles should enter the modeling domain from a boundary.

See *Also*

About the Inlet Boundary Conditions

BOUNDARY SELECTION

From the **Selection** list, choose the **Boundaries** to define the inlet.

PAIR SELECTION

If **Inlet** is selected from the **Pairs** menu, choose the pair to define. An identity pair has to be created first. Ctrl-click to deselect.

RELEASE TIMES

Enter **Release times** 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 below.

INITIAL POSITION

Select an **Initial position**—**Mesh based** (default), **Uniform distribution** (2D models) **Projected plane grid** (3D models), or **Density**.

- If **Mesh based** is selected the particles are released from a set of positions determined by a selection of geometric entities (of arbitrary dimension) in the mesh. Given a **Refinement factor** between 1 and 5, 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`.

2D

For 2D models, if **Uniform distribution** is selected, enter the **Number of particles per release** N. 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.

3D

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

- If **Density** is selected, enter a value for the **Number of particles per release** N and the **Density proportional to** ρ. Both are unitless numbers. The **Number of particles per release** is the number before Maxwellian velocity multiplication. The field **Density proportional to** is an expression - the resulting particle distribution will approximately have a density that is proportional to this expression. Evaluation of the density expression on the selected domains is taken as one value for each mesh element. The resulting distribution will look a bit random, and it will depend on the order in which the mesh elements are numbered. The distribution will probably not be exactly the same in different COMSOL versions, but the total number of particles released will always be N. The **Density proportional to** expression must be strictly positive.

INITIAL VELOCITY

Enter expressions for the **Initial particle velocity** $\mathbf{v}_0$ (SI unit: m/s) based on space dimension.

INITIAL VALUE OF AUXILIARY DEPENDENT VARIABLES

If an Auxiliary Dependent Variable has been added to the model then there is an option to enter an **Initial value of auxiliary dependent variables**. There is an option to choose a distribution function for the initial value of the auxiliary dependent variables. In this case, the number of particles simulated may increase substantially and the following options are available. The **Distribution function** option is used to select whether the initial value of the auxiliary dependent variables should be a scalar value, or a distribution function.

- When the **Distribution function** is set to **None**, enter an initial value for each of the Auxiliary Dependent Variables added to the model.
- The **Normal** option creates a normal distribution function for the initial value of the auxiliary dependent variables, with a user defined **Mean** (default 0) and **Variance** (default 1). The **Number of values** sets the number of points in the distribution function.
- The **Lognormal** option creates a log-normal distribution function for the initial value of the auxiliary dependent variables, with a user defined **Mean** (default 0)

and **Variance** (default 1). The **Number of values** sets the number of points in the distribution function.

- The **Uniform** option creates a uniform distribution function for the initial value of the auxiliary dependent variables, with a user defined **Mean** (default 0) and **Variance** (default 1). The **Number of values** sets the number of points in the distribution function.

Outlet

Use the **Outlet** feature to determine what should happen to particles when exiting the modeling domain.

See Also

- About the Outlet Boundary Conditions
- Theory for the Mathematical Particle Tracing Interface

BOUNDARY SELECTION

From the **Selection** list, choose the **Boundaries** to define the outlet.

WALL

Select a **Wall condition**—**Freeze** or **Disappear**. When the **Freeze** option 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 etc to be recovered at the instant the particles left the modeling domain. The **Disappear** option means that the particles are not visualized once they strike the outlet boundary.

Particle Continuity

Use the **Particle Continuity** feature to specify that particles should cross a pair boundary as if it were invisible.

BOUNDARY SELECTION

From the **Selection** list, choose the **Boundaries** to define the particle continuity.

PAIR SELECTION

From the **Pair Selection** list, choose the **Pairs** to define the particle continuity.

Particle Properties

Use the **Particle Properties** feature to set properties based on the **Formulation** selected on the interface settings window. Set the particle velocity for **Massless** formulations, or particle mass for **Newtonian**, **Lagrangian**, or **Hamiltonian** formulations.

See Also

For theory see:

- Newtonian Formulation
- Hamiltonian Formulation
- Lagrangian Formulation
- Massless Formulation

HAMILTONIAN

Note

This section displays for The Mathematical Particle Tracing Interface and when **Hamiltonian** is selected as the **Formulation**.

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

- If **Directly** is selected, enter a value for **Hamiltonian** H (SI unit: J). The Hamiltonian is typically a function of the particle kinetic energy and any fields which would induce a force on the particles.
- If **Manually** is selected, 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 twice as high as the **Massless**, **Lagrangian** and **Newtonian** formulations.

LAGRANGIAN

Note

This section displays 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. Much like the Hamiltonian formulation, the Lagrangian is usually a function of the particle kinetic energy and fields, which exert a force on the particles.

PARTICLE MASS

Note

This section displays when **Newtonian**, **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 in COMSOL for the electron mass, $9.10938188 \cdot 10^{-31}$kg.

PARTICLE VELOCITY

Note

This section displays 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.

Auxiliary Dependent Variable

Use the **Auxiliary Dependent Variable** feature to add additional degrees of freedom for each particle. These can be used to solve for the particle size, mass, spin, temperature, and so forth.

AUXILIARY DEPENDENT VARIABLE

Enter a **Field variable name**. The default is `rp` and can be changed to anything provided it does not conflict with the name of the variables for the position or momentum degrees of freedom. The name should not conflict with other auxiliary dependent variables. When postprocessing auxiliary dependent variables the model qualifier must be used. So, if the **Field variable name** is set to `rp` in Model 1 with tag `mod1`, then `mod1.rp` should be used.

Enter a **Source** R. There are no units assigned to the source variable because, in general, it depends on what physical quantity the auxiliary dependent variable is supposed to represent.

Under **Integrate** chose whether to integrate the equation you have defined **With respect to time** or **Along particle trajectory**. If you are interested in the residence time of a group of particles in a given system then you can set the **Source** to 1 and set **Integrate** to **With respect to time**. To compute the length of the particle trajectory you can set the **Source** to 1 and set Integrate to **Along particle trajectories**.

Release from Grid

Note

For The Charged Particle Tracing Interface and The Particle Tracing for Fluid Flow Interfaces **Newtonian** and **Massless** are available from the **Formulation** list.

For The Mathematical Particle Tracing Interface **Newtonian**, **Lagrangian**, **Massless**, and **Hamiltonian** are available from the **Formulation** list.

Use the **Release from Grid** feature to set the release times, initial position, and initial value of any auxiliary dependent variables. For **Newtonian**, **Hamiltonian** and **Lagrangian** formulations, also set the initial velocity. The sections available are based on the **Formulation** selected on the interface settings window.

RELEASE TIMES

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

INITIAL COORDINATES

Enter **Initial coordinates** based on space dimension ($q_{x,0}$, $q_{y,0}$, and $q_{z,0}$ for 3D models) for the particle positions or click the **Range** button ( ) to select and define a range of specific coordinates. There is one particle generated for each grid point in each space dimension before multiplication by Maxwellian velocity distribution (see below). The position for any particles with initial coordinates outside the geometry are set to `NaN`, so the particles do not appear during when plotted during postprocessing.

Initial Velocity

Note

For The Charged Particle Tracing Interface and The Particle Tracing for Fluid Flow Interfaces this section displays when **Newtonian** is selected as the **Formulation**.

For The Mathematical Particle Tracing Interface this section displays when **Newtonian**, **Lagrangian**, or **Hamiltonian** is selected as the **Formulation**.

Select an option from the **Initial velocity** list—**Expression** or **Maxwellian**.

- If **Expression** is selected, enter coordinates for the **Initial particle velocity** $\mathbf{v}_0$ (SI unit: m/s) based on space dimension.
- If **Maxwellian** is selected, enter the **Number of particles per velocity space direction** N_v. The default is 50. Then enter the **Temperature** T_0 (SI unit: K). The default is 293.15 K.

Note

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

- If **Constant speed, spherical** is selected, enter the **Speed** v (SI unit: m/s) and the **Number of particles in velocity space** N_vel. This option generates N_vel particles for each release point, each with the same speed. The velocity distribution is spherical for 3D models and circular for 2D.
- If **Constant speed, hemispherical** is selected, enter the **Speed** v (SI unit: m/s), the **Number of particles in velocity space** N_vel and a direction vector for the **Hemisphere axis**. This option generates N_vel particles for each release point, each with the same speed. The velocity distribution is hemispherical for 3D models and a half circle for 2D. The hemisphere axis sets the direction of the north pole of the hemisphere or half circle.

Initial Value of Auxiliary Dependent Variables

If an Auxiliary Dependent Variable has been added to the model then there is an option to enter an **Initial value of auxiliary dependent variables**. There is an option to choose a distribution function for the initial value of the auxiliary dependent variables.

In this case, the number of particles simulated may increase substantially and the following options are available. The **Distribution function** option is used to select whether the initial value of the auxiliary dependent variables should be a scalar value, or a distribution function.

- When the **Distribution function** is set to **None**, enter an initial value for each of the Auxiliary Dependent Variables added to the model.
- The **Normal** option creates a normal distribution function for the initial value of the auxiliary dependent variables, with a user defined **Mean** (default 0) and **Variance** (default 1). The **Number of values** sets the number of points in the distribution function.
- The **Lognormal** option creates a log-normal distribution function for the initial value of the auxiliary dependent variables, with a user defined **Mean** (default 0) and **Variance** (default 1). The **Number of values** sets the number of points in the distribution function.
- The **Uniform** option creates a uniform distribution function for the initial value of the auxiliary dependent variables, with a user defined **Mean** (default 0) and **Variance** (default 1). The **Number of values** sets the number of points in the distribution function.

Theory for the Mathematical Particle Tracing Interface

The equations solved by the Mathematical Particle Tracing interface depends on the formulation selected. In the following, the particle has a position vector **q** and a momentum vector **p**.

The Mathematical Particle Tracing Interface theory is described in this section:

- Newtonian Formulation
- Hamiltonian Formulation
- Lagrangian Formulation
- Massless Formulation
- Initial Conditions—Position
- Initial Conditions—Velocity
- Auxiliary Dependent Variables
- About the Boundary Conditions for the Mathematical Particle Tracing Interface

Newtonian Formulation

The Mathematical Particle Tracing interface makes it possible to solve arbitrary ordinary differential equations. The particle position is always computed using Newton's second law:

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

where m_p is the particle mass (SI unit: kg) and **F** is the force exerted on the particle (SI unit: N). The force is either specified directly or using a susceptibility multiplied by a suitable field:

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

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.

If you use a relativistic correction of the particle mass for particles with very high velocity, the particle mass m_p is computed as

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

where $\mathbf{v}$ is the particle velocity and c is the speed of light.

Hamiltonian Formulation

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

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

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

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

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

and

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

Lagrangian Formulation

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

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

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

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

where N is a matrix with elements given by:

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

Massless Formulation

The pathline of a massless particle is given by:

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

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

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

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 and Lagrangian formulations the following condition is implemented:

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

In this case every particle gets the same initial velocity.

MAXWELLIAN

When a Maxwellian initial velocity distribution is selected, an array of N_v initial velocities is created for each particle in each velocity direction. This substantially increases the number of degrees of freedom in the problem because N_v differential equations need to be solved for each initial particle position and velocity dimension. For a given initial temperature, in each velocity direction, the distribution function is given by:

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

So the total distribution function is:

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

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

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

This can clearly generate a large number of particles because the total number of initial velocities are N_v^{nsdim}. This option is useful in plasma and AC/DC modeling but not particularly useful for particle tracing in fluids.

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:

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

where θ goes from 0 to 2π and φ goes from 0 to π.

CONSTANT SPEED, HEMISPHERICAL

The constant speed, hemispherical option is the same as above, except that in 2D θ goes from 0 to π and in 3D φ goes from 0 to π/2.

Particle-Particle Interactions

COULOMB FORCE

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

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

where e is the elementary charge (SI unit: C), $k_e = 8.9875\cdot10^9$ N m^2/C^2 is a multiplication factor, $\mathbf{r}_i$ is the position vector of the i^{th} particle (SI unit: m), and $\mathbf{r}_j$ is the position vector of the j^{th} particle (SI unit: m). The Coulomb force is repulsive with 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: N·m), and σ is the collision diameter (SI unit: m).

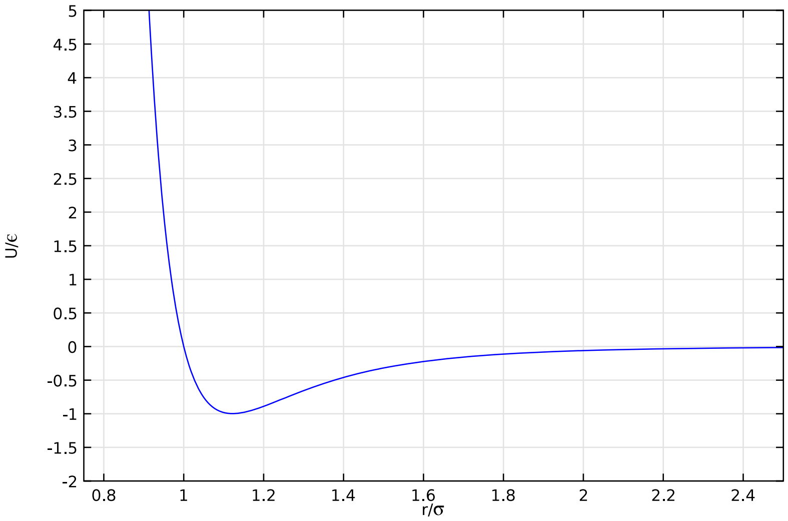


Figure 3-1: Plot of the Lennard-Jones potential normalized to the interaction strength versus the nondimensional radial distance.

Using the fact that the force is computed as:

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

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

$$\mathbf{F}_i = \frac{24\varepsilon}{\sigma}\sum_{j=1}^{N}\left[2\left(\frac{\sigma}{|\mathbf{r}_i-\mathbf{r}_j|}\right)^{13}-\left(\frac{\sigma}{|\mathbf{r}_i-\mathbf{r}_j|}\right)^{7}\right]\left(\frac{\mathbf{r}_i-\mathbf{r}_j}{|\mathbf{r}_i-\mathbf{r}_j|}\right) \tag{3-1}$$

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 3-2.

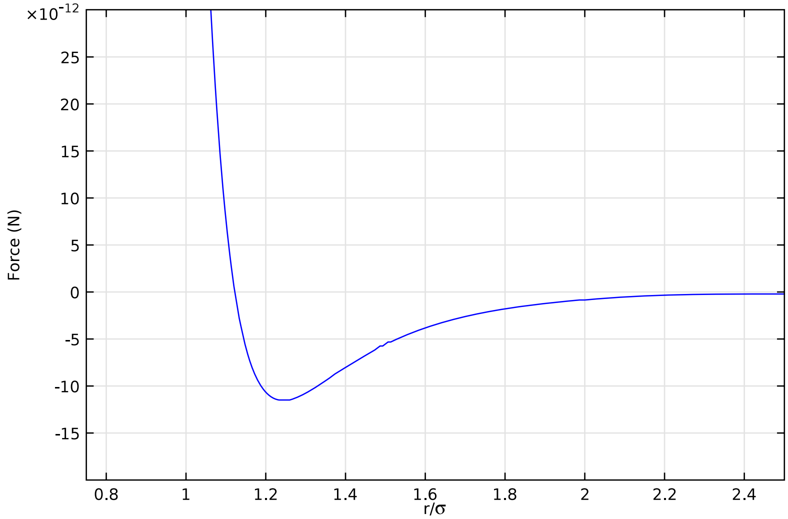


Figure 3-2: Plot of the Lennard-Jones force versus nondimensional radial distance between particles.

Auxiliary Dependent Variables

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

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

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

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

where s is the direction tangential to the motion of the particle.

About the Boundary Conditions for the Mathematical Particle Tracing Interface

The equations of motion for the particle trajectories are supplemented with a variety of options to describe how the particles behave at walls, inlets, and outlets.

ABOUT THE WALL BOUNDARY CONDITIONS

The following nomenclature is used with time as an example:

- Value of a variable pre-contact, t.
- Value of a variable at the moment of contact, t_c.
- Value of a variable post-contact, t'.

When a particle comes in contact with a wall, at time t', the following options are available and described next: disappear, freeze, stick, and bounce.

Disappear

When the particle strikes the wall it disappears from view. Mathematically the particle location is set to NaN at all timesteps after the particle makes contact with the wall:

$$\mathbf{x}' = \mathrm{NaN}$$

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

Freeze

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

$$\mathbf{x}' = \mathbf{x}_\mathrm{c}$$

$$\mathbf{v}' = \mathbf{v}_\mathrm{c}$$

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:

$$\mathbf{x}' = \mathbf{x}_\mathrm{w}$$

$$\mathbf{v}' = \mathbf{v}_\mathrm{w}$$

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:

$$\mathbf{x}' = \mathbf{x}$$

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

This conserves energy if the Hamiltonian depends isotropically on momentum.

ABOUT THE INLET BOUNDARY CONDITIONS

At the inlet the number of particles, particle position, initial velocity, and the number of releases is specified.

ABOUT THE OUTLET BOUNDARY CONDITIONS

At the outlets the particles either freeze or disappear.

4

Charged Particle Tracing

This chapter describes the Charged Particle Tracing interface found under the **AC/DC** branch (⚡) in the **Model Wizard**.

In this chapter:

The Charged Particle Tracing Interface

The **Charged Particle Tracing** interface ( ) is found under the **AC/DC** branch ( ) in the **Model Wizard**, and adds forces to ions and electrons. There are predefined forces for the elastic collision force, the electric force, and magnetic force. In addition, you can use it to model two-way coupling between the particles and fields.

When this interface is added, these default nodes are also added to the **Model Builder**—**Wall** and **Particle Properties**. Right-click the **Charged Particle Tracing** node to add other features.

Model

If you have the AC/DC Module:

- Magnetic Lens: Model Library path **ACDC_Module/Particle_Tracing/magnetic_lens**
- Quadrupole Mass Filter: Model Library path **ACDC_Module/Particle_Tracing/quadrupole_mass_filter**

If you have the Plasma Module:

- Ion Energy Distribution Function: Model Library path **Plasma_Module/Inductively_Coupled_Plasmas/ion_energy_distribution_function**

INTERFACE IDENTIFIER

The interface identifier is a text string that can be used to reference the respective physics interface if appropriate. Such situations could occur when coupling this interface to another physics interface, or when trying to identify and use variables defined by this physics interface, which is used to reach the fields and variables in expressions, for example. It can be changed to any unique string in the **Identifier** field.

The default identifier (for the first interface in the model) is `cpt`.

DOMAIN SELECTION

The default setting is to include **All domains** in the model to define the dependent variables and the equations. To choose specific domains, select **Manual** from the **Selection** list.

PARTICLE PROPERTIES

Select a **Formulation**—**Newtonian** or **Massless**. This selection changes the available sections displayed on the Particle Properties and Release settings windows. If **Newtonian** is selected, a Force node can also be added to the model.

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 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.

If the **Secondary emission** option is checked in the **Wall** boundary condition, there is the potential to generate a large number of secondary particles. The **Maximum number of secondary particles** prevents an inordinate number of particles from being generated by capping them at the number supplied in the edit field.

The **Relativistic correction** option, applicable for particles with very high speed, means you can take relativistic effects on the particle mass into account. Select **On** from the **Relativistic correction** list (the default is to not use a relativistic correction). The particle mass m_p is then computed as

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

where $\mathbf{v}$ is the particle velocity and c is the speed of light.

INCLUDE OUT-OF-PLANE DEGREES OF FREEDOM

2D *2D Axi*	For 2D and 2D axisymmetric models, select whether to **Include out-of-plane degrees of freedom** in the model. This option adds extra degrees of freedom for the out-of-plane position and velocity. From the **Include out-of-plane degrees of freedom** list, select **On** or **Off** (the default).

When **On** is selected, additional degrees of freedom are added for each particle for the out-of-plane position and velocity. This option is primarily intended for 2D and 2D axially symmetric models, where 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.

DEPENDENT VARIABLES

The dependent variables (field variables) are the **Particle position** and **Particle position components**. The name can be changed but the names of fields and dependent variables must be unique within a model.

See Also

- Show More Physics Options
- Domain, Boundary, Pair, and Global Features 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 Features for the Charged Particle Tracing Interface

The Charged Particle Tracing Interface has these domain, boundary, pair, and global features available and described in this section:

- Elastic Collision Force
- Electric Force
- Magnetic Force
- Particle Field Interaction

These features are described for the **Mathematical Particle Tracing** interface (listed in alphabetical order):

- Auxiliary Dependent Variable
- Force
- Inlet
- Outlet
- Particle Continuity
- Particle-Particle Interaction
- Particle Properties (default)
- Release
- Release from Grid

- Wall (the default boundary condition)

See Also

The Weak Contribution Node in the *COMSOL Multiphysics User's Guide*

Particle Properties

Use the **Particle Properties** feature to specify the particle mass and charge number when using the Newtonian formulation. Specify the particle velocity when using the **Massless** formulation.

PARTICLE MASS

Note

This section displays when **Newtonian** is selected as the **Formulation**.

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

CHARGE NUMBER

Note

This section displays when **Newtonian** is selected as the **Formulation**.

Enter a value for the **Charge number** Z (dimensionless). The default is –1 corresponding to the charge number of an electron.

PARTICLE VELOCITY

Note

This section displays 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.

Electric Force

Note

The **Electric Force** node is also only available when **Newtonian** is selected as the **Formulation** on the The Charged Particle Tracing Interface settings window.

Use the **Electric Force** node to define the electric part of the Lorentz force. The particles are accelerated in the same orientation as the electric field. 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.

DOMAIN SELECTION

From the **Selection** list, choose the domains for the electric force.

ELECTRIC FORCE

Select an option from the **Specify force using** list—**Electric field** or **Electric potential**.

- If **Electric field** is selected, 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 **Electric field** list.
- If **Electric potential** is selected, 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 **Electric potential** list.

ADVANCED SETTINGS

Select the **Multiply force by phase angle** check box in cases where the field is computed in the frequency domain. When the particle trajectories are computed, the value of the field used to compute the force is computed from the complex field via:

$$\mathbf{E} = \mathrm{real}(\tilde{\mathbf{E}}\exp(j\omega t))$$

where $\mathbf{E}$ is the electric field value used to compute the force, $\tilde{\mathbf{E}}$ is the complex valued electric field and ω is the angular frequency.

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.

Magnetic Force

Note

The **Magnetic Force** node is also only available when **Newtonian** is selected as the **Formulation** on the The Charged Particle Tracing Interface settings window.

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. The force is specified via a magnetic flux density.

DOMAIN SELECTION

From the **Selection** list, choose the domains for the magnetic force.

MAGNETIC FORCE

Enter values or expressions in the table for the **Magnetic flux density** $\mathbf{B}$ (SI unit: T) based on space dimension, or, if the magnetic flux is computed by another physics interface, select a **Magnetic flux density** from the **Magnetic flux density** list.

ADVANCED SETTINGS

Select the **Multiply force by phase angle** check box in cases where the field is computed in the frequency domain. When the particle trajectories are computed, the value of the field used to compute the force is computed from the complex field via:

$$\mathbf{B} = \mathrm{real}(\tilde{\mathbf{B}}\exp(j\omega t))$$

where $\mathbf{B}$ is the magnetic flux density value used to compute the force, $\tilde{\mathbf{B}}$ is the complex valued magnetic flux density, and ω is the angular frequency.

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.

Elastic Collision Force

Note

The **Elastic Collisions** node is only available when **Newtonian** is selected as the **Formulation** on the The Charged Particle Tracing Interface settings window.

Use the **Elastic Collisions** node to model cases where the charged particles undergo elastic collisions with the background gas. An elastic collision means there is no change in the chemical energy of the colliding particles. At very low pressures (sub millitorr) elastic collisions can usually be neglected. As the background pressure increases, collisions become more and more important because they can inhibit the particles from acquiring high energies.

DOMAIN SELECTION

From the **Selection** list, choose the domains for the elastic collision.

ELASTIC COLLISION

Enter a value for the **Collision frequency** v (SI unit: Hz). In a typical case this is proportional to the background number density, the collision cross section and the velocity of the particle.

Enter coordinates for the **Velocity field** $\mathbf{u}$ (SI unit: m/s) based on space dimension. To model the case of a stagnant background fluid, simply leave all the velocity components as zero.

Particle Field Interaction

Use the **Particle Field Interaction** node if space charge effects are important. This node adds two-way coupling between the particles and field. That is, the field exerts a force on the particles and the particles exert a space charge on the field. In order to see this effect it is necessary to solve for the particles and fields simultaneously.

DOMAIN SELECTION

From the **Selection** list, choose the domains for the particle field interaction.

PARTICLE FIELD INTERACTION

Enter values or expressions for the **Charge multiplication factor** n (dimensionless) and **Electric potential** V (SI unit: V). The electric potential is usually computed by an Electrostatics or Electric Currents interface. The **Charge multiplication factor** can be used to represent each particle as a group of n particles. This means that the magnitude of the space charge exserted on the field by each particle is multiplied by n. This prevents having to model an overwhelming number of particles to capture space charge effects.

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
- Particle Properties Theory
- Electric Force Theory
- Magnetic Force Theory
- Elastic Collisional Force Theory
- Particle Field Interaction Theory

Introduction to the Charged Particle Tracing Interface Theory

Motion of charged particles in electromagnetic field is best understood by starting with the Lagrangian for a charge in an electromagnetic field:

$$L = -m_p c^2 \sqrt{1-\mathbf{v}\cdot\mathbf{v}/c^2} + q\mathbf{A}\cdot\mathbf{v} - qV$$

where m_p is the particle mass, c is the speed of light, $\mathbf{v}$ is the particle velocity, q is the particle charge, $\mathbf{A}$ is the magnetic vector potential, and V is the electric scalar potential. The particle charge is:

$$q = Ze$$

where Z is the charge number and e is the elementary charge (SI unit: C). For low velocities, after subtracting the rest energy, the Lagrangian becomes:

$$L = \frac{m_p v^2}{2} + q\mathbf{A}\cdot\mathbf{v} - qV$$

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 $\mathbf{q}$ is the particle position vector. The right-hand side of the Lagrange equation is:

$$\frac{\partial L}{\partial \mathbf{q}} = \nabla L = q(\nabla(\mathbf{A}\cdot\mathbf{v}) - \nabla V)$$

Using some elementary vector calculus, this can be rewritten as:

$$\frac{\partial L}{\partial \mathbf{q}} = q(\mathbf{v}\cdot\nabla)\mathbf{A} + q(\mathbf{v}\times\nabla\times\mathbf{A}) - q\nabla V$$

So, the Lagrange equation becomes:

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

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

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

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

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

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 can be written as:

$$\frac{d}{dt}(m\mathbf{v}) = -q\mathbf{E} + q(\mathbf{v}\times\mathbf{B}) \qquad (4\text{-}4)$$

The term on the right-hand side of Equation 4-4 is called the Lorentz force. In addition to forces due to an electromagnetic field, the particles undergo collisions with a background fluid. The elastic collisional damping force with a background gas is approximated by the expression:

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

where $\mathbf{u}$ is the mass average velocity of the background gas (SI unit: m/s) and ν is the collision frequency (SI unit: 1/s). For a constant collision cross section, the friction factor is:

$$\nu = N\sigma|\mathbf{v}|$$

where N is the number density of the background gas (SI unit: 1/m^3) and σ is the collision cross section (SI unit: m^2). Note that this acts as a continuous force on the particle rather than changing the particle velocity direction when a collision event occurs. Such collisions require a probabilistic modeling approach, which is not supported in this version of the Particle Tracing Module.

Particle Properties Theory

The Particle Properties feature adds the following equation contribution:

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

where $\mathbf{F}_t$ is zero in the absence of any force features. Adding additional forces to the model add contributions to $\mathbf{F}_t$. You specify the particle mass m and particle charge q in this feature.

Electric Force Theory

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

$$\mathbf{F}_t = q\mathbf{E}$$

When an electric potential is used to specify the electric field, the following is used:

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

When the electric field is computed in the frequency domain it is complex valued. The field must be cast into a real value which depends on the angular frequency and the simulation time:

$$\mathbf{E}(t) = \mathrm{real}(\tilde{\mathbf{E}}\exp(j\omega t))$$

where $\tilde{\mathbf{E}}$ is the complex valued electric field, ω is the angular frequency, and t is time.

Magnetic Force Theory

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

$$\mathbf{F}_t = q(\mathbf{v} \times \mathbf{B})$$

When the magnetic flux density is computed in the frequency domain it is complex valued. The field must be cast into a real value which depends on the angular frequency and the simulation time:

$$\mathbf{B}(t) = \mathrm{real}(\tilde{\mathbf{B}}\exp(j\omega t))$$

where $\tilde{\mathbf{B}}$ is the complex valued electric field, ω is the angular frequency, and t is time

Elastic Collisional Force Theory

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

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

Note

The limitations of this feature are described in the Introduction to the Charged Particle Tracing Interface Theory section.

Particle Field Interaction Theory

The Particle Field Interaction feature adds a contribution to the equation governing the field rather than the particle. The equation contribution acts like a space charge density point source at the particle coordinates:

$$\rho = enZ\sum_{i=1}^{N}\delta(q_{x,i}, q_{y,i}, q_{z,i})$$

where ρ is the space charge density, e is the elementary charge, n is the charge multiplication factor, Z is the charge number, δ is the delta function, and q is the particle coordinate. The sum is over all particles.

5

Particle Tracing for Fluid Flow

This chapter describes the Particle Tracing for Fluid Flow interface found under the **Fluid Flow** branch (≋) in the **Model Wizard**.

In this chapter:

The Particle Tracing for Fluid Flow Interface

The **Particle Tracing for Fluid Flow** interface ( ), found under the **Fluid Flow** branch ( ) in the **Model Wizard**, computes 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 also be added. It is also possible to compute the particle mass and temperature as well as particle-fluid interactions.

Model

- Particle Trajectories in a Laminar Static Mixer: Model Library path **Particle_Tracing_Module/Fluid_Flow/laminar_mixer_particle**
- If you have the CFD Module, Particle Tracing in a Micromixer: Model Library path **CFD_Module/Particle_Tracing/micromixer_particle_tracing**
- If you have the Acoustics Module, Acoustic Levitator: Model Library path **Acoustics_Module/Tutorial_Models/acoustic_levitator**

When this interface is added, these default nodes are also added to the **Model Builder**—**Wall** and **Particle Properties**. Right-click the **Particle Tracing for Fluid Flow** node to add other features.

INTERFACE IDENTIFIER

The interface identifier is a text string that can be used to reference the respective physics interface if appropriate. Such situations could occur when coupling this interface to another physics interface, or when trying to identify and use variables defined by this physics interface, which is used to reach the fields and variables in expressions, for example. It can be changed to any unique string in the **Identifier** field.

The default identifier (for the first interface in the model) is `fpt`.

DOMAIN SELECTION

The default setting is to include **All domains** in the model to define the dependent variables and the equations. To choose specific domains, select **Manual** from the **Selection** list.

PARTICLE PROPERTIES

Select a **Formulation**—**Newtonian** or **Massless**. This selection changes the available sections displayed on the **Particle Properties** and **Release** settings windows. If **Newtonian** is selected, a **Force** node can also be added to the model.

Select a **Wall accuracy order**. This 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 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.

ADVANCED SETTINGS

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.

Select the **Compute particle mass** check box to compute particle masses (the default is to not compute particle masses). When this option is activated an additional equation for the particle mass is computed by solving an additional ordinary differential equation per particle.

If the **Secondary emission** check box is selected in the **Wall** boundary condition, there is the potential to generate a large number of secondary particles. The **Maximum number of secondary particles** prevents an inordinate number of particles from being generated by capping them at the number supplied in the edit field (default: 10,000 particles).

INCLUDE OUT-OF-PLANE DEGREES OF FREEDOM

2D

2D Axi

For 2D and 2D axisymmetric models, select whether to **Include out-of-plane degrees of freedom** in the model. This option adds extra degrees of freedom for the out-of-plane position and velocity.

From the **Include out-of-plane degrees of freedom** list, select **On** or **Off** (the default).

When **On** is selected, additional degrees of freedom are added for each particle for the out-of-plane position and velocity. This option is primarily intended for 2D and 2D axially symmetric models, where 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.

DEPENDENT VARIABLES

The dependent variables (field variables) are the **Particle position**, **Particle position components**, **Particle temperature**, and **Particle mass**. The name can be changed but the names of fields and dependent variables must be unique within a model.

See Also

- Show More Physics Options
- Domain, Boundary, Pair, and Global Features for the Particle Tracing for Fluid Flow Interface
- Theory for the Mathematical Particle Tracing Interface
- Newtonian Formulation
- Massless Formulation
- Theory for the Particle Tracing for Fluid Flow Interface

Domain, Boundary, Pair, and Global Features for the Particle Tracing for Fluid Flow Interface

The Particle Tracing for Fluid Flow Interface has these domain, boundary, pair, and global features available.

These features are described in this section (listed in alphabetical order):

- Acoustophoretic Force
- Brownian Force
- Dielectrophoretic Force
- Drag Force
- Electric Force
- Fluid-Particle Interaction
- Gravity Force
- Magnetic Force

- Magnetophoretic Force
- Thermophoretic Force

See Also

For a theoretical background to the forces, see Theory for the Particle Tracing for Fluid Flow Interface.

These features are described for the Mathematical Particle Tracing interface (listed in alphabetical order):

- Auxiliary Dependent Variable
- Force
- Inlet
- Outlet
- Particle Continuity
- Particle-Particle Interaction
- Particle Properties
- Release
- Release from Grid
- Wall (the default boundary condition)

See Also

The Weak Contribution Node in the *COMSOL Multiphysics User's Guide*

Drag Force

Note

The **Drag Force** node is only available when **Newtonian** is selected as the **Formulation** on the The Particle Tracing for Fluid Flow Interface settings window.

Use the **Drag Force** node to define the turbulent dispersion, velocity field, dynamic viscosity, and turbulent kinetic energy.

DOMAIN SELECTION

From the **Selection** list, choose the domains for the drag force.

DRAG FORCE

Select the **Turbulent dispersion** check box to activate random motion of the particles due to the fact that a turbulent flow field is present.

For the **Drag Law**, select between **Stokes** (default) and **Schiller-Naumann**. Stokes drag is applicable for particles which have a relative Reynolds number much less than one. This should be a good enough approximation in most cases. If the particle Reynolds number is greater than one then the Schiller-Naumann option should be used.

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

If the **Schiller-Naumann** option is selected for the **Drag Law** then enter a value or expression for the fluid density.

Enter a value or expression for the **Dynamic viscosity** μ (SI unit: Pa·s) and **Turbulent kinetic energy** k (SI unit: $\mathrm{m^2/s^2}$). If a physics interface is present which computes the turbulent kinetic energy, it can be selected directly from the list.

See Also

Particle Motion in a Fluid in the theory section.

Brownian Force

Note

The **Brownian Force** node is only available when **Newtonian** is selected as the **Formulation** on the The Particle Tracing for Fluid Flow Interface settings window.

Use the **Brownian Force** node to account for diffusion of suspended particles in a fluid according to Einstein's theory. When using this feature, a random force is applied on each particle at each time step. This often forces the solver to take very small time steps because the default relative and absolute tolerances solver settings are quite strict (10^{-5} and 10^{-6}, respectively). In order to make the model solve efficiently, it may be necessary to relax these tolerances. The solver needs to take enough time steps so that the random force is applied enough times to be meaningful, but not so many time steps that the model takes a very long time to solve. As a general rule of thumb, it is usually best to relax both the tolerances to 10^{-3} and then perform a convergence study on the tolerances.

DOMAIN SELECTION

From the **Selection** list, choose the domains for the Brownian force.

BROWNIAN FORCE

To defined the Brownian force, there are two physical properties that you need to define as inputs for the force expression: the fluid temperature and dynamic viscosity.

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

Enter a value or expression for the **Dynamic viscosity** 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 it can come from the selected material.

ADVANCED SETTINGS

The random number generator used in COMSOL 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. 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 In the **Model Builder**, right-click **Global Definitions** (≡) and select **Parameters** (Pi).

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`.

See Also

- Global Parameters in the *COMSOL Multiphysics User's Guide*
- Parametric Sweep in the *COMSOL Multiphysics Reference Guide*
- Additional Forces in the theory section.

Gravity Force

Note

The **Gravity Force** node is only available when **Newtonian** is selected as the **Formulation** on the The Particle Tracing for Fluid Flow Interface settings window.

Use the **Gravity Force** node to define the gravity vector and density. The gravity force causes particles with a suitably high mass and density to move in the same direction as the orientation of the gravity vector. For submicron particles, the drag and other external forces can dominate and gravity can have little effect on the particle trajectories.

DOMAIN SELECTION

From the **Selection** list, choose the domains for the gravity force.

GRAVITY FORCE

Enter coordinates based on space dimension for the **Gravity vector g** (SI unit: $\mathrm{m/s^2}$) and **Density** ρ (SI unit: $\mathrm{kg/m^3}$). The default magnitude for the gravity vector is `g_const`, which is a built-in physical constant corresponding to the standard acceleration due to gravity on Earth.

2D

The default direction for the gravity vector is in the y-direction for 2D models.

2D Axi

3D

The default direction for the gravity vector is in the z-direction for 2D axisymmetric and 3D model.

The fluid density can come either from a material, from a physics interface which defines or computes the density, or be a user-defined expression. The default behavior is to take the density **From material**.

See Also

Particle Motion in a Fluid in the theory section.

Acoustophoretic Force

Note

The **Acoustophoretic Force** node is only available when **Newtonian** is selected as the **Formulation** on the The Particle Tracing for Fluid Flow Interface settings window.

Use the **Acoustophoretic Force** node to add forces on small particles due to acoustic radiation. This force only adds equation contributions if the acoustic pressure was computed from a **Frequency Domain Study** type. This is because the magnitude of the acoustophoretic force depends on the angular frequency.

DOMAIN SELECTION

From the **Selection** list, choose the domains for the acoustophoretic force.

ACOUSTIC PROPERTIES

Enter a value or expression for the following:

- **Pressure** p (SI unit: Pa). This should be computed from another physics interface using a **Frequency Domain Study Type**.

- **Speed of sound** c (SI unit: m/s) of the surrounding fluid. This can either be taken from a material or user defined.
- **Density** ρ (SI unit: kg/m^3) of the surrounding fluid. This can either be taken from a material or user defined.

ACOUSTOPHORETIC FORCE

Select an option from the **Specify** list—**Particle bulk modulus**, **Particle compressibility**, or **Compressional speed of sound**. Then, if required change the defaults and enter a value or expression the available fields as follows:

- **Particle bulk modulus** K_p (SI unit: Pa). The default is 2.2 GPa ($2.2{\cdot}10^9$).
- **Particle compressibility** β_p (SI unit: 1/Pa). The default is $1/(2.2{\cdot}10^9)$ 1/Pa.
- **Compressional speed of sound** c_p (SI unit: m/s). The default is 3000 m/s.

ADVANCED SETTINGS

Select the **Use piecewise polynomial recovery on field** check box to smooth the radiation pressure using piecewise polynomial recovery. This can give a much more accurate representation of the radiation pressure because it uses information on adjacent mesh elements to reconstruct the field. If a coarse mesh is used to compute the field then this option can be especially useful.

See Also

- Frequency Domain in the *COMSOL Multiphysics Reference Guide*
- Solver Studies and Study Types in the *COMSOL Multiphysics User's Guide*
- Additional Forces in the theory section.

Electric Force

Note

The **Electric Force** node is only available when **Newtonian** is selected as the **Formulation** on the The Particle Tracing for Fluid Flow Interface settings window.

Use the **Electric Force** node to add 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.

DOMAIN SELECTION

From the **Selection** list, choose the domains for the electric force.

ELECTRIC FORCE

Select an option from the **Specify force using** list—**Electric field** or **Electric potential**.

- If **Electric field** is selected, 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 **Electric field** list.
- If **Electric potential** is selected, 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 **Electric potential** list.

ADVANCED SETTINGS

Select the **Multiply force by phase angle** check box in cases where the field is computed in the frequency domain. When the particle trajectories are computed, the value of the field used to compute the force is computed from the complex field via:

$$\mathbf{E} = \mathrm{real}(\tilde{\mathbf{E}}\exp(j\omega t))$$

where $\mathbf{E}$ is the electric field value used to compute the force, $\tilde{\mathbf{E}}$ is the complex-valued electric field, and ω is the angular frequency.

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.

See Also

Additional Forces in the theory section.

Magnetic Force

Note

The **Magnetic Force** node is only available when **Newtonian** is selected as the **Formulation** on the The Particle Tracing for Fluid Flow Interface settings window.

Use the **Magnetic Force** node to add 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.

DOMAIN SELECTION

From the **Selection** list, choose the domains for the magnetic force.

MAGNETIC FORCE

Enter values or expressions in the table for the **Magnetic flux density** $\mathbf{B}$ (SI unit: T) based on space dimension, or, if the magnetic flux is computed by another physics interface, select a **Magnetic flux density** from the **Magnetic flux density** list.

ADVANCED SETTINGS

Select the **Multiply force by phase angle** check box in cases where the field is computed in the frequency domain. When the particle trajectories are computed, the value of the field used to compute the force is computed from the complex field via:

$$\mathbf{B} = \mathrm{real}(\tilde{\mathbf{B}}\exp(j\omega t))$$

where $\mathbf{B}$ is the magnetic flux density value used to compute the force, $\tilde{\mathbf{B}}$ is the complex-valued magnetic flux density, and ω is the angular frequency.

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.

See Also

Additional Forces in the theory section.

Dielectrophoretic Force

Note

The **Dielectrophoretic Force** node is only available when **Newtonian** is selected as the **Formulation** on the The Particle Tracing for Fluid Flow Interface settings window.

Use the **Dielectrophoretic Force** node to add a dielectrophoretic force to the particles. The force is specified via an electric potential or the electric field. The force has a different meaning depending on whether the source electric field is computed in a **Stationary** or **Frequency Domain** study.

Additionally, piecewise polynomial recovery can be used which can give a more accurate representation of the specified electric field. The influence of the dielectrophoretic force on the particles depends on the difference in permittivity between the particles and the fluid. When the particle relative permittivity is greater than the relative permittivity of the fluid, the particles are attracted to regions where the absolute electric field is strong. When the particle relative permittivity is less than the relative permittivity of the fluid, the particles are attracted to regions where the absolute electric field is weak.

DOMAIN SELECTION

From the **Selection** list, choose the domains for the dielectrophoretic force.

DIELECTROPHORETIC FORCE

Select an option from the **Specify force using** list—**Electric field** or **Electric potential**.

- If **Electric field** is selected, 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 **Electric field** list.
- If **Electric potential** is selected, 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 **Electric potential** list.

PARTICLE PROPERTIES

Enter a value or expression for the **Particle relative permittivity** (SI unit: dimensionless). The value must be greater than or equal to one.

FLUID PROPERTIES

Enter a value or expression for the **Fluid relative permittivity** (SI unit: 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 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.

See Also

- Frequency Domain and Stationary in the *COMSOL Multiphysics Reference Guide*
- Solver Studies and Study Types in the *COMSOL Multiphysics User's Guide*
- Additional Forces in the theory section.

Magnetophoretic Force

Note

The **Magnetophoretic Force** node is only available when **Newtonian** is selected as the **Formulation** on the The Particle Tracing for Fluid Flow Interface settings window.

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.

DOMAIN SELECTION

From the **Selection** list, choose the domains for the magnetophoretic force.

MAGNETOPHORETIC FORCE

Enter a value or expression for the magnetic field. If the field is computed by another physics interface then it can be selected from the **Magnetic Field** list.

PARTICLE PROPERTIES

Enter a value or expression for the **Particle relative permeability** (SI unit: dimensionless). The value must be greater than or equal to one.

FLUID PROPERTIES

Enter a value or expression for the **Fluid relative permeability** (SI unit: 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.

See Also	Additional Forces in the theory section.

Thermophoretic Force

Note	The **Thermophoretic Force** node is only available when **Newtonian** is selected as the **Formulation** on the The Particle Tracing for Fluid Flow Interface settings window.

Use the **Thermophoretic Force** node to include forces on particles due to a temperature gradient in the background fluid. The force is specified via the background temperature and other properties of the fluid.

Additionally, piecewise polynomial recovery can be used which can give a more accurate representation of the specified temperature.

DOMAIN SELECTION

From the **Selection** list, choose the domains for the thermophoretic force.

THERMOPHORETIC FORCE

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

Enter a value or expression for the **Dynamic viscosity** of the fluid (SI unit: Pa·s). The dynamic viscosity can also come from a material or it can be computed by another physics interface.

Enter a value or expression for the **Density** of the fluid (SI unit: kg/m^3). The density can also come from a material or it can be computed by another physics interface.

Enter a value or expression for the **Thermal Conductivity** of the fluid (SI unit: W/m K). The thermal conductivity can also come from a material.

PARTICLE PROPERTIES

Enter a value or expression for the **Particle thermal conductivity** (SI unit: W/m K).

Enter a value or expression for the **Thermophoretic correction factor** (SI unit: dimensionless). The default value is 1.17.

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.

See Also

Additional Forces in the theory section.

Fluid-Particle Interaction

Note

The **Fluid-Particle Interaction** node is also only available when **Newtonian** is selected as the **Formulation** on the The Particle Tracing for Fluid Flow Interface settings window.

Use the **Fluid-Particle Interaction** node to define the velocity field of the fluid. This feature adds additional contributions to the equations governing the fluid flow. The particle degrees of freedom must be computed simultaneously with the fluid flow degrees of freedom in order to model this effect.

DOMAIN SELECTION

From the **Selection** list, choose the domains for the fluid particle interaction.

FLUID-PARTICLE INTERACTION

Enter coordinates for the **Velocity field** $\mathbf{u}$ (SI unit: m/s) based on space dimension. In order to capture the two-way coupling between the particles and velocity field, the velocity field variables must be dependent variables. Typically this would come from a **Laminar Flow** physics interface for single-phase fluid flow.

See Also	Theory for the Particle Tracing for Fluid Flow Interface

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
- Additional Forces
- Fluid-Particle Interaction
- References for the Particle Tracing for Fluid Flow Interface

Particle Motion in a Fluid

The particle momentum comes from Newton's second law, which states that the net force on a particle is equal to its time rate of change of its linear momentum in an inertial reference frame:

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

DRAG FORCE

In the above equation, $\mathbf{F}_D$ is the Drag Force is defined as:

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

where

- m_p is the particle mass (SI unit: kg)
- τ_p is the particle velocity response time (SI unit: s)
- $\mathbf{v}$ is the velocity of the particle (SI unit: m/s)
- $\mathbf{u}$ is the fluid velocity (SI unit: m/s)
- $\mathbf{F}_g$ is the gravitational force vector (SI unit: N), and
- $\mathbf{F}_{\mathrm{ext}}$ is any other external force (SI unit: N).

The particle velocity response time for spherical particles in a laminar flow is defined as:

$$\tau_p = \frac{\rho_p d_p^2}{18\mu}$$

where 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).

When the drag force is set to Schiller Naumann, the particle velocity response time is redefined as:

$$\tau_p = \frac{4\rho_p d_p^2}{3\mu C_D \mathrm{Re}_r}$$

where:

$$C_D = \frac{24}{\mathrm{Re}_r}(1 + 0{,}15\mathrm{Re}_r^{0{,}637})$$

and

$$\mathrm{Re}_r = \frac{\rho\|\mathbf{u} - \mathbf{v}\|d_p}{\mu}.$$

When turbulent dispersion is activated, the fluid velocity used in the drag force becomes:

$$\mathbf{u} = \mathbf{U} + \mathbf{u}'$$

where $\mathbf{U}$ is the mean velocity and $\mathbf{u}'$ is a the turbulent fluctuation, defined as:

$$\mathbf{u}' = \zeta\sqrt{\frac{2k}{3}}$$

where k is the turbulent kinetic energy. The quantity ζ is a normally distributed random number with zero mean and unit standard deviation.

GRAVITY FORCE

The Gravity Force is given by:

$$\mathbf{F}_g = m_p\mathbf{g}\frac{(\rho_p - \rho)}{\rho_p}$$

where ρ is the density of the surrounding fluid (SI unit: kg/m^3) and $\mathbf{g}$ is the gravity vector.

Additional Forces

In addition to the Drag Force and Gravity Force described in Particle Motion in a Fluid, the following predefined forces may be added to influence particle motion.

- Brownian Force
- Electric Force
- Magnetic Force
- Dielectrophoretic Force
- Acoustophoretic Force
- Thermophoretic Force
- Magnetophoretic Force

BROWNIAN FORCE

The Brownian Force feature adds the following contribution to $\mathbf{F}_{\mathrm{ext}}$:

$$\mathbf{F}_{\mathrm{ext}} = \zeta\sqrt{\frac{12\pi k_B \mu T r_p}{\Delta t}} \qquad (5\text{-}1)$$

where

- Δt is the time step taken by the solver
- r_p is the particle radius
- T is the absolute fluid temperature
- η is the fluid dynamic viscosity,
- k_B is Boltzmann's constant
- μ is the fluid viscosity, and
- ζ is a normally distributed random number with a mean of zero and unit standard variation.

  As explained in Ref. 5, 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.

ELECTRIC FORCE

The Electric Force feature adds the following contribution to $\mathbf{F}_{\mathrm{ext}}$:

$$\mathbf{F}_{\mathrm{ext}} = eZ\mathbf{E}$$

where e is the elementary charge (SI unit: s A), Z is the charge number (SI unit: dimensionless), and $\mathbf{E}$ is the electric field (SI unit: V/m). The electric field can be given directly or computed from an electric potential:

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

When the electric field is computed in the frequency domain it is complex valued. The field must be cast into a real value that depends on the angular frequency and the simulation time:

$$\mathbf{E}(t) = \mathrm{real}(\tilde{\mathbf{E}}\exp(j\omega t))$$

where $\tilde{\mathbf{E}}$ is the complex-valued electric field, ω is the angular frequency, and t is time.

MAGNETIC FORCE

The Magnetic Force feature adds the following contribution to $\mathbf{F}_{\mathrm{ext}}$:

$$\mathbf{F}_{\mathrm{ext}} = eZ(\mathbf{v}\times\mathbf{B})$$

When the magnetic flux density is computed in the frequency domain it is complex valued. The field must be cast into a real value that depends on the angular frequency and the simulation time:

$$\mathbf{B}(t) = \mathrm{real}(\tilde{\mathbf{B}}\exp(j\omega t))$$

where $\tilde{\mathbf{B}}$ is the complex-valued electric field, ω is the angular frequency, and t is time.

DIELECTROPHORETIC FORCE

The Dielectrophoretic Force feature adds the following contribution to $\mathbf{F}_{\mathrm{ext}}$ for the case of a static electric field:

$$\mathbf{F}_{\mathrm{ext}} = 2\pi r_p^3\varepsilon_0\varepsilon_f\left(\frac{\varepsilon_p-\varepsilon_f}{\varepsilon_p-2\varepsilon_f}\right)\nabla|\mathbf{E}|^2$$

where ε_f is the relative permittivity of the fluid, ε_p is the relative permittivity of the particle, and $\mathbf{E}$ is the electric field. In the case that the electric field is computed in the frequency domain, the following is added:

$$\mathbf{F}_{\mathrm{ext}} = 2\pi r_p^3\varepsilon_0\mathrm{real}(\varepsilon_f)\mathrm{real}\left(\frac{\varepsilon_p-\varepsilon_f}{\varepsilon_p-2\varepsilon_f}\right)\nabla|\mathbf{E}_{\mathrm{rms}}|^2$$

where $\mathbf{E}_{\mathrm{rms}}$ denotes the root mean square electric field.

ACOUSTOPHORETIC FORCE

The Acoustophoretic Force is a special subclass of the CFD forces, which are the acoustic radiation forces on small particles. The present theory is derived for inviscid fluids and small particles. This implies that the expressions are valid for particles of diameter $d_p = 2r$ much smaller than the acoustic wavelength λ and of size larger than the acoustic boundary layer thickness λ_{bl}. The last condition is not necessary for buoyant particles (of same density ρ_p as the fluid density ρ_0). The present expressions are based on the work of Gorkov (Ref. 3) which is reviewed by Bruus (see Ref. 4).

Following the references the radiation force on a particle is given as

$$\mathbf{F}_{\mathrm{aco}} = -\nabla U^{\mathrm{rad}}$$

where U^{rad} is the radiation potential given by

$$U^{\mathrm{rad}} = V_p\left[f_1\frac{1}{2\rho_0 c_0^2}\langle p_{\mathrm{in}}^2\rangle - f_2\frac{3}{4}\rho_0\langle v_{\mathrm{in}}^2\rangle\right]$$

$$f_1 = 1-\frac{\rho_0 c_0^2}{\rho_p c_p^2} = 1-\frac{\beta_p}{\beta_0} = 1-\frac{K_0}{K_p}$$

$$f_2 = \frac{2(\rho_p-\rho_0)}{2\rho_p+\rho_0}$$

where p_{in} is the incident (external) pressure field and v_{in} is the incident particle velocity amplitude, c_0 is the speed of sound in the fluid, c_{p} is the compressional speed of sound in the particle, and β_p is the compressibility of the particle (reciprocal of bulk modulus = $(K_{\mathrm{p}})^{-1}$). The $\langle\ \rangle$ terms are time averaged over one period.

In the case of an acoustic field derived from a time-harmonic analysis (frequency-domain analysis) where the dependent variables are complex valued, the time-averaged expressions simplify as

$$\langle p_{\mathrm{in}}^2\rangle = \frac{1}{2}(p_{\mathrm{in}}p^*{}_{\mathrm{in}})$$

$$\langle v_{\mathrm{in}}^2\rangle = \frac{1}{2}(v_{\mathrm{in}}v^*{}_{\mathrm{in}}) = \frac{1}{2}(\mathbf{v}_{\mathrm{in}}\cdot\mathbf{v}^*{}_{\mathrm{in}})$$

where * is the complex conjugate. For the case of pressure acoustics the velocity is simply given as

$$\mathbf{v}_{\text{in}} = \frac{-1}{i\omega\rho_0}\nabla p$$

Thus the present acoustic radiation forces are only space dependent and not time dependent as they are time averaged.

THERMOPHORETIC FORCE

The Thermophoretic Force accounts for a force on a particle due to gradients in the temperature of the background fluid. The thermophoretic force is defined as:

$$\mathbf{F}_{tp} = \frac{6\pi d_p \mu^2 C_s (k_f/k_p) \nabla T}{\rho(2(k_f/k_p)+1)T}$$

where k_f is the thermal conductivity of the fluid, k_p is the particle thermal conductivity, T is the fluid temperature, ρ is the fluid density, and C_s is a constant equal to 1.17. 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.

MAGNETOPHORETIC FORCE

The Magnetophoretic Force causes particle motion away from regions where the magnetic field is strong. 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:

$$\mathbf{F}_m = 2\pi r_p^{\ 3} \mu_0 \mu_{r,f} K \nabla \mathbf{H}^2$$

where $\mathbf{H}$ is the magnetic field, $\mu_{r,f}$ is the fluid relative permeability, $\mu_{r,p}$ is the particle relative permeability, and K is defined as:

$$K = \frac{\mu_{r,p} - \mu_{r,f}}{\mu_{r,p} + 2\mu_{r,f}}$$

The force is only valid for non-conducting particles, so additional force contributions due to eddy currents induced in particles are not taken into account.

Fluid-Particle Interaction

When a particle experiences a drag force due to the fluid, the particle in turn exerts a momentum force on the fluid. Because the position of the particles at any given instant is between the mesh nodes, the exact momentum force imposed on the fluid cannot be implemented. Instead, the net momentum force imposed on the fluid, $\mathbf{F}_{\text{fluid}}$, is smeared over the mesh element corresponding to the particles location. The momentum force on the fluid is:

$$\mathbf{F}_{\text{fluid}} = \frac{\mathbf{F}_D}{\text{dvol}}$$

where `dvol` is the mesh element volume.

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. L.P. Gorkov, *Sov. Phys. Doklady*, vol. 6, p. 773, 1962.

4. H. Bruus, Lecture Notes: Microfluidics and ultrasound acoustophoreses. http://www.nanotech.dtu.dk/Research/Research%20groups/TMF.aspx

5. M. Kim and A.L. Zydney, “Effect of Electrostatic, Hydrodynamic, and Brownian Forces on Particle Trajectories and Sieving in Normal Flow Filtration,” *Journal of Colloid and Interface Science*, vol. 269, pp. 425–431, 2004.

6

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, please see the glossary in the *COMSOL Multiphysics User's Guide*. For references to more information about a term, see the index.

Glossary of Terms

Brownian motion The random drifting of particles suspended in a fluid.

Coulomb force The force between charged particles which is inversely proportional to the square of the distance between the particles.

Dispersed flow A fluid-particle system where particle-fluid and particle-particle interactions need to be accounted for in the model.

Hamiltonian A convenient way of describing how a system of particles interact with surrounding fields. The Hamiltonian is usually defined as the sum of the kinetic and potential energy.

Lagrangian A convenient way of describing how a system of particles interact with surrounding fields. The Lagrangian is usually defined as the kinetic energy minus the potential energy.

Maxwellian velocity distribution Describes the probability that a velocity is near a given value as a function of the temperature of the system.

Residence time The average amount of time that a particle spends in a particular system. The residence time can be computed by adding Auxiliary dependent variables.

Space charge effects When the number density of charged particles is sufficiently high, they can affect the field in which they are placed. This is often referred to as a "space charge effect."

Sparse flow A particle laden flow is described as a sparse flow when the particles have no appreciable effect on the motion of the fluid.

Transmission probability The probability that a particle will transmit from a given selected boundary or domain to another.

Index